

MacTech®

The Journal of Macintosh Technology and Development

Using Rendezvous

How to discover and publish
services on the LAN

by Wolfgang Ante, Vienna, Austria.

Reviews in this Issue:

- **Book Review: MAC OS X ADVANCED DEVELOPMENT TECHNIQUES**
- **Movie Review: REVOLUTION OS**
- **FUGU: SECURE FILE TRANSFERS WITH FUGU**
- **TiGlide AND iGlide**
- **MindVision FileSTORM**

\$8.95 US
\$12.95 Canada
ISSN 1067-8360
Printed in U.S.A.



UNFAIR



Qt/Mac, the Mac OS X version of the Qt multiplatform C++ application framework, provides a uniquely unfair technological advantage in software development. Qt/Mac lets Mac developers develop in C++ on the platform they love most, while targeting additional operating systems, such as Windows or Linux. Qt/Mac also offers:

- fully object-oriented, elegant API
- single-source, multiplatform development solution
- native compiling and applications that run at native speed



TROLLTECH™

- integration with Project Builder, allowing developers to use Apple's own IDE for editing, compiling and debugging Qt/Mac applications
- OpenGL support, for sophisticated 3D graphics on the Mac
- native look & feel, such as the Aqua style

Don't take our word for it. Visit us at www.trolltech.com/mac. Look at our list of Fortune 500 customers. Read what programmers say about Qt. Download the evaluation version, and see how easy C++ development can be.

Qt /MAC: THE UNFAIR ADVANTAGE IN C++ DEVELOPMENT

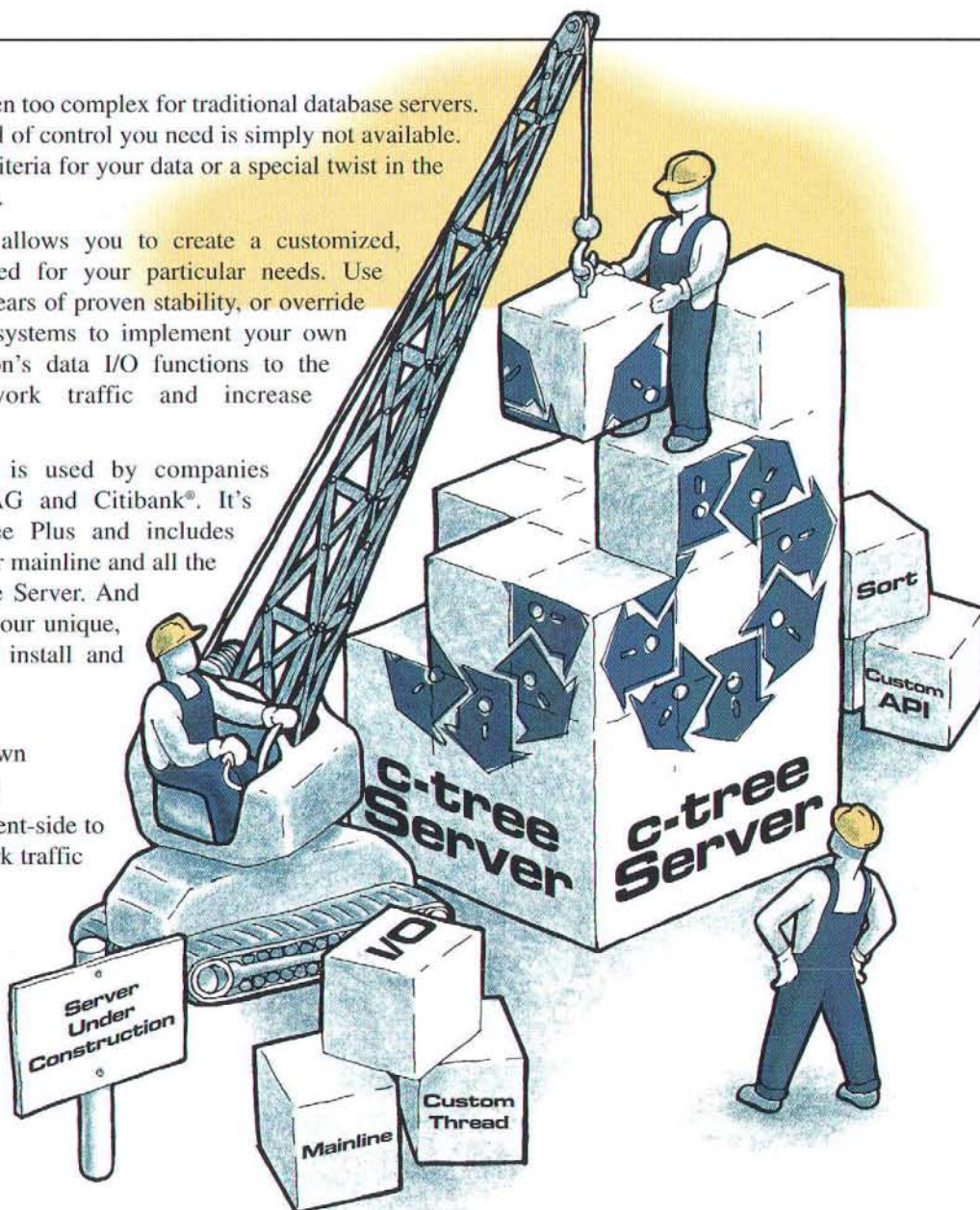
CUSTOMIZE YOUR DATABASE SERVER WITH THE C-TREE® SERVER SDK

Today's database demands are often too complex for traditional database servers. The functionality and precise level of control you need is simply not available. Perhaps you need alternate sort criteria for your data or a special twist in the threading or communication logic.

FairCom's c-tree® Server SDK allows you to create a customized, industrial-strength server designed for your particular needs. Use FairCom's kernel, with over 20 years of proven stability, or override functionality within specific subsystems to implement your own subtleties. Move your application's data I/O functions to the server-side to decrease network traffic and increase performance!

FairCom's c-tree Server SDK is used by companies worldwide such as Software AG and Citibank®. It's integrated seamlessly into c-tree Plus and includes complete source code to the server mainline and all the interface subsystems to the c-tree Server. And best of all, once you've created your unique, customized server, it is easy to install and administer: no DBA required!

- Enhance our server with your own custom server-side functionality
- Move functionality from the client-side to the server-side to reduce network traffic and increase performance
- Modify or replace entire server subsystems
- Complete source for the server mainline, key server subsystems, and client-side
- Flexible OEM licensing



Visit www.faircom.com/ep/mt/sdk today to take control of your server!




FairCom®
www.faircom.com

USA	573.445.6833
EUROPE	+39.035.773.464
JAPAN	+81.59.229.7504
BRAZIL	+55.11.3872.9802

DBMS Since 1979 • 800.234.8180 • info@faircom.com

Other company and product names are registered trademarks or trademarks of their respective owners.

© 2002 FairCom Corporation



Adaptive Server®
Enterprise 12.5
MAC OS X

HELPING FILEMAKER PRO CUSTOMERS SCALE TO NEW HEIGHTS.

Want to enhance the performance of FileMaker Pro on MAC OS X? The same database engine that runs Wall Street can help you do just that.

ASE 12.5 increases the reliability, availability and scalability of your FileMaker Pro application. It provides better data integrity, world-class security and the ability to handle thousands of transactions per minute. It'll also give your users the power of SQL queries.

ASE 12.5 for MAC OS X is yet one more example of how

Sybase is helping today's leading companies achieve Information Liquidity: a highly profitable state where all of your information is transformed into real economic value.

A FREE Developer's Edition download is available online at sybase.com/filemaker.

INFORMATION LIQUIDITY.



BETTER WHEN EVERYTHING WORKS TOGETHER.™

How To Communicate With Us

In this electronic age, the art of communication has become both easier and more complicated. Is it any surprise that we prefer e-mail?

If you have any questions, feel free to call us at 805/494-9797 or fax us at 805/494-9798.

If you would like a subscription or need customer service, feel free to contact Developer Depot Customer Service at 877-MACTECH

DEPARTMENTS

Orders, Circulation, & Customer Service

Press Releases

Ad Sales

Editorial

Programmer's Challenge

Online Support

Accounting

Marketing

General

Web Site (articles, info, URLs and more...)

E-Mail/URL

cust_service@mactech.com

press_releases@mactech.com

ad_sales@mactech.com

editorial@mactech.com

prog_challenge@mactech.com

online@mactech.com

accounting@mactech.com

marketing@mactech.com

info@mactech.com

http://www.mactech.com

The MacTech Editorial Staff

Publisher • Neil Ticktin

Editor-in-Chief • Dave Mark

Managing Editor • Jessica Stubblefield

Regular Columnists

Getting Started

by Dave Mark

QuickTime ToolKit

by Tim Monroe

Mac OS X Programming Secrets

by Scott Knaster

Reviews/KoolTools

by Michael R. Harvey

Patch Panel

by John C. Welch

Section 7

by Rich Morin

Untangling the Web

by Kevin Hemenway

John and Pals' Puzzle Page

by John Vink

Regular Contributors

Vicki Brown, Erick Tejkowski,
Paul E. Seving

MacTech's Board of Advisors

Jordan Dea-Mattson, Jim Straus
and Jon Wiederspan

MacTech's Contributing Editors

- Michael Brian Bentley
- Vicki Brown
- Marshall Clow
- John. C. Daub
- Tom Djajadiningrat
- Bill Doernfeld, Blueworld
- Andrew S. Downs
- Richard V. Ford, Packeteer
- Gordon Garb, Sun
- Ilene Hoffman
- Chris Kilbourn, Digital Forest
- Rich Morin
- John O'Fallon, Maxum Development
- Will Porter
- Avi Rappoport, Search Tools Consulting
- Ty Shipman, Kagi
- Chuck Shotton, BIAP Systems
- Cal Simone, Main Event Software
- Steve Sisak, Codewell Corporation
- Chuck Von Rospach, Pfaidworks

Xplain Corporation Staff

Chief Executive Officer • Neil Ticktin

President • Andrea J. Sniderman

Controller • Michael Friedman

Production Manager • Jessica Stubblefield

Production/Layout • Darryl Smart

Marketing Manager • Nick DeMello

Account Executive • Lorin Rivers
adsales@mactech.com • 800-5-MACDEV

Events Manager • Susan M. Worley

International • Rose Kemps

Network Administrator • David Breffitt

Accounting • Jan Webber, Marcie Moriarty

Customer Relations • Laura Lane, Susan Pomrantz

Shipping/Receiving • Joel Licardie

Board of Advisors • Steven Geller, Blake Park,
and Alan Carsrud

All contents are Copyright 1984-2003 by Xplain Corporation. All rights reserved. MacTech and Developer Depot are registered trademarks of Xplain Corporation. RadGad, Useful Gifts and Gadgets, Xplain, DevDepot, Depot, The Depot, Depot Store, Video Depot, Movie Depot, Palm Depot, Game Depot, Flashlight Depot, Explain It, MacDev-1, THINK Reference, NetProfessional, NetProLive, JavaTech, WebTech, BeTech, LinuxTech, MacTech Central and the MacTutorMan are trademarks or service marks of Xplain Corporation. Sprocket is a registered trademark of eSprocket Corporation. Other trademarks and copyrights appearing in this printing or software remain the property of their respective holders.

MacTech Magazine (ISSN: 1067-8360 / USPS: 010-227) is published monthly by Xplain Corporation, 850-P Hampshire Road, Westlake Village, CA 91361-2800. Voice: 805/494-9797, FAX: 805/494-9798. Domestic subscription rates are \$47.00 per year. Canadian subscriptions are \$59.00 per year. All other international subscriptions are \$97.00 per year. Domestic source code disk subscriptions are \$77 per year. All international disk subscriptions are \$97.00 a year. Please remit in U.S. funds only. Periodical postage is paid at Thousand Oaks, CA and at additional mailing office.

POSTMASTER: Send address changes to **MacTech Magazine**, P.O. Box 5200, Westlake Village, CA 91359-5200.

COVER STORY

16 Using Rendezvous

How to discover and publish services on the LAN

by Wolfgang Ante, Vienna, Austria.

QUICKTIME

32 From Algorithm to Animation: the Sequel

Making QuickTime movies which illustrate abstract processes

By Jay Martin Anderson

TIMBUKTU PRO 6.0.3 FOR OS X REVIEWED

44 From Here to Timbuktu

A venerable tool is updated for OS X

By Chris Kilbourn

GETTING STARTED

6 TinyViewer, Part Deux

By Dave Mark, Editor In Chief

QUICKTIME TOOLKIT

48 Studio 54

Developing QuickTime Applications with AppleScript Studio

by Tim Monroe

REVIEWS

14 Book Review: Mac OS X Advanced Development Techniques

By Ron Davis

30 Movie Review: Revolution OS

The rise of the free software and open source movements

By Andrew S. Downs

38 Fugu: Secure File Transfers With Fugu

Review and how to for Fugu

By Brian Shin

42 TiGlide and iGlide

Eliminate the squeak, level the screen

By Michael R. Harvey

58 MindVision FileStorm

Creating Mac OS X installers with FileStorm

By Miro Jurisic

MAC OS X PROGRAMMING SECRETS

72 Panther and G5: The Future is Almost Now

By Scott Knaster

UNTANGLING THE WEB

64 User, Meet Apache. Apache, Hug.

Or: How To Learn More About Your Cuddly New Web Server.

By Kevin Hemenway, Questionable Intender

Copyright 2003 by Dave Mark

Getting Started: TinyViewer, Part Deux

Last month's column was a *lot* of fun. So much so, that I was really torn on what to do in this month's column. I really want to continue the iPod notes exploration. And I've been playing with these cool AppleScripts that I'm just dying to write about. Then there's TinyViewer, the app from my May column. I *really* want to finish playing with Interface Builder and updating the TinyViewer interface. So that's what I'm gonna do this month and we'll start on the other stuff next month.

TINYVIEWER RECAP

In last month's column, we used Interface Builder to build the beginnings of an interface for TinyViewer, our Cocoa update of the old PictViewer app. We used Project Builder to create a *Cocoa Application* project, then double-clicked on the *MainMenu.nib* file to open the nib file using Interface Builder.

We went through the four interface windows that appear – the app's main window, the menu bar window, the palette window, and the window labeled *MainMenu.nib*, which is a central repository for all the resources and instances from your project.

We dragged out an *NSImageView* from the palette window into the main app window, then used the dashed, blue lines to align and resize the *NSImageView* inside the window. We selected *Show Info* from the *Tools* menu and used the *Info* window to customize our *NSImageView* so it was *Editable* and *Enabled*. Making it *Editable* was especially important to our project as it allowed us to drag-and-drop an image on the window.

We also used the *Info* window to set the *Autosizing* settings, so when we resized the window, the image resized in a pleasing way.

Finally, we did the coolest thing of all. We used Interface Builder to take the interface for a spin by selecting *Test Interface* from the *File* menu. I absolutely love this part of

Interface Builder. The idea of prototyping an app, then test driving the interface, all without writing a single line of code, is amazingly useful. But beyond that, I love the fact that you can archive complex object hierarchies inside a .nib file, then bring them to life inside a completely different application. Think about this. You design and test the TinyViewer interface inside Interface Builder, then save the .nib file, wheel over to Project Builder, build and run the app, and the exact same interface appears. Again, all without writing a single line of code.

Think back to your ResEdit days. You could edit the elements that made up the interface, even do a bit of stringing them together (though not much), but when it came time to bring them to life, you had to write a *lot* of code. Even though PowerPlant allowed you to create all the usual interface elements and nest them within nestable views, you still had to write the code that instantiated all your objects and brought them to life.

Interface Builder brings this process to a new level. And I like it.

NOW WHAT DO WE DO?

OK, so what's next? Take a look at **Figure 1**. This is the menu that appears when you ask Project Builder to build and run the TinyViewer application. A couple of notes here. First, the most eagle-eyed among you might have spotted a slight discrepancy between the title of this menu and what you see on your screen. Nicely done. When you run your app inside of Interface Builder, Interface Builder actually breathes life into your objects and yields control. In effect, Interface Builder turns into your running app, until you quit. You'll know this is happening cause your app's menu title will actually say Interface Builder.

Dave Mark is a long-time Mac developer and MacTech contributor. Random facts about Dave? He is addicted to *Age of Empires*. He can't seem to stop camping. And his iPod has him hypnotized with its shiny silver grooviness.

32

< 1 year
2 years >

\$64



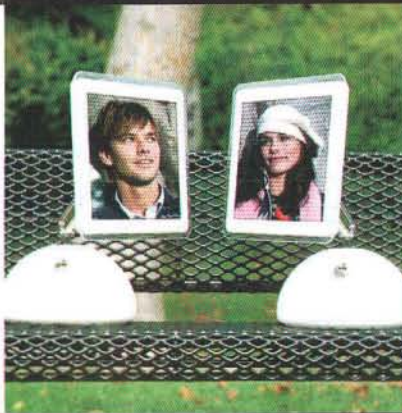
INTERVIEWS

TAPPING INTO THE WORLD OF CELEBRITIES AND THEIR MACS, ONLY MACDIRECTORY OFFERS EXCLUSIVE INTERVIEWS. GET A CLOSE AND PERSONAL VIEW FROM SARAH JESSICA PARKER, STING, STEVE JOBS, MADONNA, HARRY CONNICK JR., GEORGE LUCAS, JENNIFER JASON LEIGH, STEVE WOZ AND OTHER LEADERS IN THE MAC COMMUNITY.



FEATURES

DESIGNERS, WRITERS, MUSICIANS, BUSINESS LEADERS & OUR TECHNICAL EXPERT TEAM OFFER THEIR OWN PERSONAL INTERPRETATION OF THINGS THAT ONLY THE MAC SYSTEM CAN DELIVER. FEATURING OVER 240 PAGES OF REVIEWS, INTERVIEWS, NEWS, INSIGHTS, TRENDS AND THE LARGEST MACINTOSH BUYERS' GUIDE INCLUDING OVER 5,000 MAC PRODUCTS AND SERVICES.



CULTURE

MACDIRECTORY TAKES YOU TO THE WILDEST CORNERS OF THE WORLD AND UNCOVERS HOW MACINTOSH COMPUTERS ARE BEING USED BY OTHER CULTURES. TRAVEL TO JAPAN, AUSTRALIA, GERMANY, BRAZIL, INDIA, RUSSIA & LEARN MORE ABOUT APPLE'S CULTURAL IMPACT AROUND THE GLOBE.

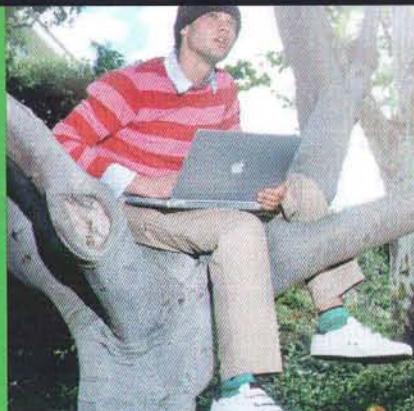
MacDirectory

BEYOND ANY MACINTOSH MAGAZINE. SUBSCRIBE.

< Subscribe

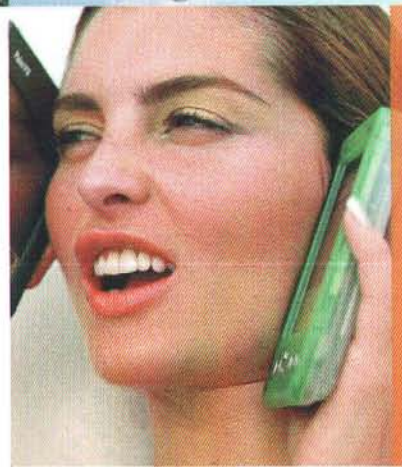
www.macdirectory.com/mw.html

SEND CHECK OR MONEY ORDER TO:
MACDIRECTORY SUB DEPT.
326 A STREET, 2C
SOUTH BOSTON, MA 02110



REVIEWS

FIND OUT ALL YOU NEED TO KNOW ABOUT THE LATEST MAC PRODUCTS INCLUDING THE HOTTEST MAC OS SOFTWARE AND HARDWARE.



WIN!

SUBSCRIBE TO MACDIRECTORY AND YOU WILL AUTOMATICALLY ENTER OUR SWEEPSTAKES FOR A CHANCE TO WIN A NEW TITANIUM!



Figure 1. TinyViewer's main menu.

When you run your app in Project Builder, Project Builder actually launches your app. As you can see in **Figure 1**, this version of TinyViewer features a TinyViewer menu, riddled with the generic name *NewApplication*. Let's change this menu to something a little more appropriate.

Note that when you run your app under Interface Builder, the IB dock icon changes into a cool, single-pole, double-throw switch (ya know, like the one Gene Wilder threw to bring life to *Young Frankenstein*). Quit the app and the old familiar IB icon returns. Since your app is not running as a separate process, there's no separate dock icon for your app.

But when ProjectBuilder runs your app, you'll find your app represented by an elegantly designed dock icon distinct and separate from the Project Builder icon (and by elegant, I mean generic and boring, unless of course you've designed one yourself).

Changing the TinyViewer Menu

Quit your app if it is running, then (if you are not already in Interface Builder) double-click *MainMenu.nib* in the project window (Files tab, TinyViewer/Resources) to open the .nib file in Interface Builder. Find the *MainMenu.nib* main window (see **Figure 2**) and double-click on the *MainMenu* icon.

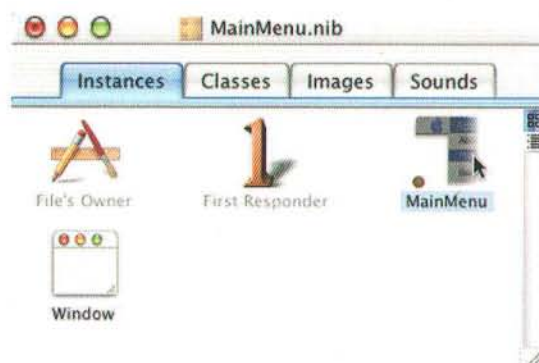


Figure 2. The *MainMenu.nib* window, with *MainMenu* selected.

When the *MainMenu* window appears, start by double-clicking the left -most menu title, the one that says *NewApplication*, and changing it to say *TinyViewer*.

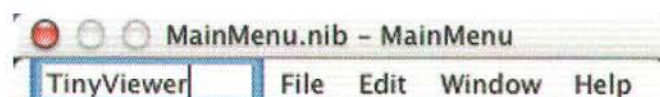


Figure 3. Editing the *TinyViewer* menu title.

Next, click on the menu title *TinyViewer*. A menu should appear. Find the three occurrences of *NewApplication* and change them all to say *TinyViewer* (see **Figure 4**). Make the same change to the single occurrence of *NewApplication* in the *Help* menu.

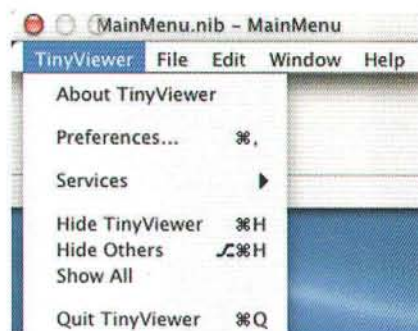


Figure 4. The complete *TinyViewer* menu.

Change the Window Title

Next step: close the menu window and click on the *TinyViewer* window to bring it to the front. Now select *Show Info* from the *Tools* menu to bring up the inspector window. If you did this correctly, the inspector window will be titled *NSWindow Info*. Make sure *Attributes* is selected from the popup menu and change the window's title to *TinyViewer*. As soon as you hit enter or click out of the field, the *TinyViewer* window's title should change to *TinyViewer*.

Very important! Be sure to save your .nib file in Interface Builder if you want to go back to Project Builder and test your changes.

Figure 5 shows my copy of *TinyViewer* with a picture dragged in and the Window's title changed. In a future column, we're going to add some code to the project to change the window title to the file name of the drag-and-dropped image. Any idea how to do this? One solution is to subclass *NSImageView* and insert your own dragging completion code. We'll also add in an *NSScrollView* to make the *NSImageView* scrollable. This one is doable completely within Interface Builder if you want to take a crack at this yourself.

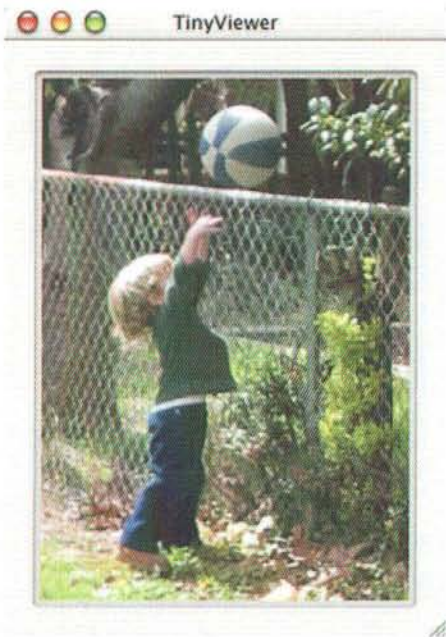


Figure 5. *TinyViewer in action. That's Ryan getting rid of my ball.*

Create the About Panel

In the olden days of Mac development, the About Box was a modal dialog box with some text in it. Over time, people started customizing these dialogs and, eventually, replaced the canned About Box with customized windows. Some (Adobe PhotoShop, for example) even went so far as to write their own custom window procedures, leading to odd shaped windows, sometimes with holes in them.

Over time, the "official" method of producing About Boxes evolved, moving from About Boxes to About Windows to the current model, the About Panel. There's a terrific article on cocoadevcentral.com by Brock Brandeberg, that'll take you on a whirlwind tour of the modern About Panel:

<http://cocoadevcentral.com/articles/000071.php>

I'll give you the short version here, but definitely take a look at Brock's article. Well worth the read.

Just as the original Toolbox contained a canned routine for putting up an About Box, Cocoa's *NSApplication* class contains a member called *orderFrontStandardAboutPanel:*

```
- (void)orderFrontStandardAboutPanel:(id)sender;
```

This function is called automatically when *About TinyViewer* is selected from the *TinyViewer* menu (obviously, substitute your app name for *TinyViewer*). *OrderFrontStandardAboutPanel:* looks in your application's bundle for a set of elements it uses to build the About Panel. Alternatively, you can use the function *orderFrontStandardAboutPanelWithOptions:* to roll your own About Panel at run time:



Fetch

Good doggie.

X

Fetchsoftworks.com

Version 4.0.3 now available.

```

- (void)orderFrontStandardAboutPanelWithOptions:
    (NSDictionary *)optionsDictionary:

```

This comment is from *NSApplication.h*. It describes the *optionsDictionary* parameter, but is useful in understanding what elements you can add to your application's bundle (more on this below):

```

/* Optional keys in optionsDictionary:

```

```

@"Credits": NSAttributedString displayed in the info area of
the panel. If not specified, contents obtained from
"Credits.rtf" in [NSBundle mainBundle];
if not available, blank.

```

```

@"ApplicationName": NSString displayed in place of the
default app name. If not specified, uses the value of
CFBundleName (localizable). Fallback is [[NSProcessInfo
processInfo] processName].

```

```

@"ApplicationIcon": NSImage displayed in place of
NSApplicationIcon. If not specified, use [NSImage
imageNamed:@"NSApplicationIcon"]; if not available, generic
icon.

```

```

@"Copyright": NSString containing the copyright string. If
not specified, obtain from the value of
NSHumanReadableCopyright (localizable) in infoDictionary; if
not available, leave blank.

```

```

@"Version": NSString containing the build version number of
the application ("58.4", "1.2d3"); displayed as "Version
58.4" or "1.0 (v58.4)" depending on the presence of
ApplicationVersion. If not specified, obtain from the
CFBundleVersion key in infoDictionary; if not specified or
empty string, leave blank.

```

```

@"ApplicationVersion": NSString displayed as the application
version ("1.0", "Mac OS X", "3", "WebObjects 3.5", ...),
before the build version. If not specified, obtain from
CFBundleShortVersionString key in infoDictionary..
*/

```

For starters, go into *TextEdit* and create your own About Panel text for *TinyViewer*. Save the file as *Credits.rtf*. Note that the name is case-sensitive and has to be exactly *Credits.rtf*. The file *Credits.rtf* has been used for About Panels since the birth of Mac OS X. Recent releases of the OS also recognize *Credits.html* and *Credits.rtf*. Obviously, *Credits.html* allows you to include links in your About Panel and *Credits.rtf* allows you to use an rtf package. You can include all three files in your application's bundle, but priority is given to *Credits.html*, then *Credits.rtf*, then *Credits.rtf*.

Why have more than one? Your best bet is to include *Credits.rtf*, in case your app is used by someone with an old version of Mac OS X installed. Whether you supplement this with *Credits.rtf* or *Credits.html* is your call.

Adding Credits.rtf to the Project File

Let's add *Credits.rtf* to our project. Once you get that to work, it's an easy step to add *Credits.html* or *Credits.rtf* to the mix.

In the project window, select the *Files* tab, then make sure the *Resources* triangle is open. In the *Finder*, click on your *Credits.rtf* file and drag it into the project window, into the *Resources* area. As you drag, you'll see a bold black line appear that indicates where the file will be dropped (see **Figure 6**).

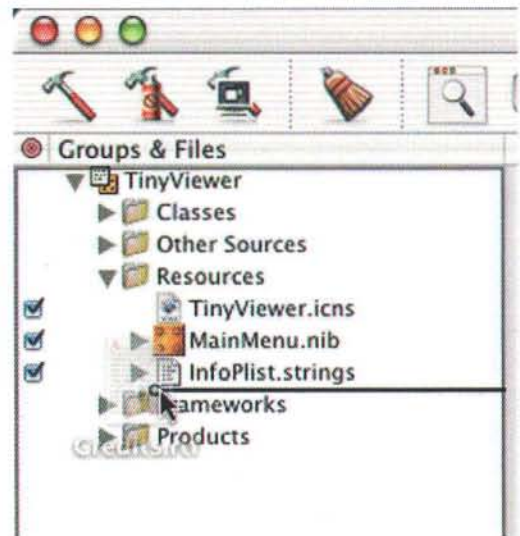


Figure 6. Dragging *Credits.rtf* into the project file.

When you release your mouse button, the dialog in **Figure 7** will appear. Be sure to check the "Copy items" checkbox at the top of the dialog. This tells Project Builder to copy the file into your project hierarchy. You can then do as you like with the original, cause Project Builder lets you edit the copy by double-clicking it.



Figure 7. The *Add File* dialog. Be sure the "Copy items" checkbox is checked.

Fill in the Bundle

My Alert Panel is shown in **Figure 8**. It is made up of a collection of resources, including an icon, a collection of info.plist strings, and the *Credits.rtf* file. You already know how to add the *Credits.rtf* file. To create your application's icons,

ching!"
"Cha-Ching!"
"Cha-Ching!"

"Cha-Ching!"

"Cha-Ching!"



New Privilege 6

Generate new software revenue streams — securely, effectively.

Selling, distributing and activating software electronically isn't new. But doing it securely AND all from one source certainly is news! With Privilege from Aladdin, you can have:

Via the Internet:

- A secure ESD process, transforming casual sharing into a super distribution channel for trialware
- Increased customer satisfaction and loyalty with short and simple purchasing process and resumable downloads
- Instant access to international markets with multi-language support

Via digital media and CD:

- Your choice of CD's, or pre-loaded trialware on new computers, with flexible software activation, try-before-you-buy, and try-only
- Capture new opportunities with full control over licensing terms, offers and the user's experience

Take advantage of every new revenue opportunity available today! Call us at 1-800-562-2543, or visit GoPrivilege.com to receive a FREE Privilege information pack.

Privilege[™]
SECURE SOFTWARE eCOMMERCE

you'll want to use a program like *PhotoShop* to create the artwork, then import that artwork into *IconComposer*. *IconComposer* is part of the developer tools and is found in */Developer/Applications/*.

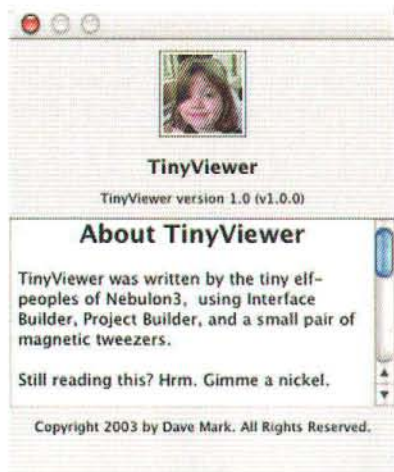


Figure 8. My TinyViewer About Panel. Hey! That's Kelley!

Building an icon is entire column in itself. The key is to make sure you create an alpha channel for the 128-by-128 version of the icon, save as a tiff, then import into *IconComposer*. If your icon looks funky when you save and reopen it, you've probably not done the alpha channel magic properly. We'll tackle this in a future column. If you want to try your hand at doing this, be sure to save the icon as a .icns file, then drag it into the project, just as you did with *Credits.rtf*. I called my icon *TinyViewer.icns*.

Your next step is to click on the *Targets* tab in the Project Window, then click on the *TinyViewer* target. This will bring up a list of settings you'll need to edit in order to fill out the rest of your About Panel, add an icon to the project, etc. **Figure 9** shows my settings.

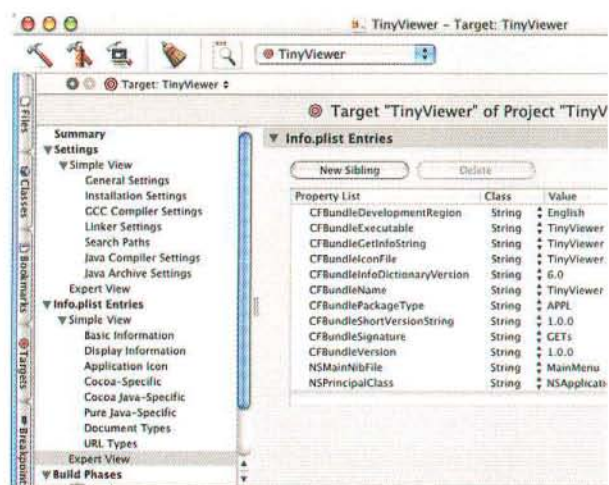


Figure 9. TinyViewer's Info.plist entries.

Note that the *CFBundleSignature* is set to *GETs*. I logged on to <http://developer.apple.com> and registered this creator code with Apple specifically for *TinyViewer*. Feel free to use this code yourself, since I'll only use it for *Getting Started* projects. We'll dig into registering creator codes in a future column.

Once those changes are made, go back to the *Files* tab and open the *Resources* triangle. Click on the *InfoPlist.strings* item. You'll see something that looks like this:

```
/* Localized versions of Info.plist keys */

CFBundleName = "TinyViewer";
CFBundleShortVersionString = "TinyViewer version 1.0";
CFBundleGetInfoString = "TinyViewer 1.0, Copyright 2003 by Dave Mark. All Rights Reserved.";
NSHumanReadableCopyright = "Copyright 2003 by Dave Mark. All Rights Reserved.";
```

I've edited my versions to reflect the settings I want for *TinyViewer*. Note that the *InfoPlist.strings* file is your localized strings file. You might contract out to a localizing firm to take your *English* plist file and translate it to a *German* plist file. You'd keep both localized files in your project and the right strings would be displayed, depending on the language setting on your computer.

Think of the *Info.plist* file as the true repository for your plist keys. Any strings that need to be localized should be kept in each localized *InfoPlist.strings* file. The localized version, if it exists, will override any matching key in the *Info.plist* file. When building your project, just do the *Targets* version first, then do the *Files* version and you should be fine.

A couple of books that really helped with this month's column: *Joe Zobkiw's excellent Mac OS X, Advanced Development Techniques* really digs deep into a wide variety of Cocoa-related topics. And, as I've mentioned in a previous column, *Bill Cheeseman's Cocoa Recipes for Mac OS X: The Vermont Recipes* is an invaluable resource. I definitely owe both these guys a beverage of their choice. And an extra tall one for Mike Trent for helping to unravel all those pesky Cocoa mysteries. Thanks!

TILL NEXT MONTH...

Man.

There was *way* more that I wanted to get into. I just ran out of space. I am just loving Project Builder, Interface Builder and Cocoa. Not sure what we'll dig into next month. There's a little more iPod madness left in me. I also just got this really good *AppleScript* book. Hrm. Maybe a column on scripting my iPod? See you soon...☺

**What does software protection
have to do with money?**

**11 billion US\$ lost worldwide
through software piracy!**

WIBU-KEY Software Protection

■ Software goes online

Electronic Software Distribution –
safely protected by WIBU-KEY.

■ Pay-Per-Use

Usage dependent accounting of
your software.

■ License Management

You can easily and flexibly create and
manage network licenses.

■ Mac OS 9 & X

WIBU-KEY supports Mac OS, Windows
and heterogeneous networks.

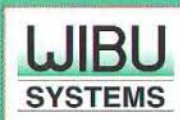
**Test the WIBU-KEY Protection Kit
free and decide for yourself!**

1-800-986-6578

sales@griftech.com



The Key is in Your Hands!



WIBU-SYSTEMS USA, Inc.
Seattle, WA 98101
email: info@wibu.com

www.griftech.com
www.wibu.com

Test Kits also available at:

Belgium wibu@impakt.be, Denmark lean@danbit.dk, Finland finebyte@finebyte.com, France info@neol.fr, Hungary info@mrsoft.hu, Japan info@suncarla.co.jp,
Jordan, Lebanon starsoft@cyberia.net.lb, Korea dhikimn@wibu.co.kr, Luxembourg wibu@impakt.be, Netherlands wibu@impakt.be,
Portugal dubit@dubit.pt, Thailand preecha@dpff-th.com, United Kingdom info@codework.com, USA sales@griftech.com

By Ron Davis

Book Review: Mac OS X Advanced Development Techniques

I have to confess that I worked with Joe Zobkiw on my first job out of college working on Virex almost ten years ago. I was one of the tech reviewers for his previous book *A Fragment of Your Imagination*, which covered the esoteric programming arts of init, cdev, and other code fragments. It was one of those books that covered stuff no other book covered, the really low level crap most of us attempt to avoid. At the time there was next to no documentation on coding these things and Zobkiw's book was written to meet that need.

Mac OS X Advanced Development Techniques (MADT) is like that book. Not that it talks about the stuff you are trying to avoid writing, but it is talking about the things no one else is. There isn't as much low level stuff you can mess with in OSX, but there are a lot of little nooks and crannies of code that need explanations. Need to write a System Preference panel? It's there. How about adding one of those system-wide menus? Got you covered. They're called Status Items, Chapter 9. Screen saver? Chapter 10.

It also covers some less stand-alone stuff, like threads, frameworks and how to talk between a Cocoa app and a command-line tool. There is also a good chapter of "snippets", little pieces of code you often want, but that don't merit a long discussion.

SECTION 1: OVERVIEW

In my opinion, this book is really aimed at the experienced OSX developer. The introduction says anyone who can compile a program in Project Builder can use it. I think you probably need a little more experience than that. I was interested in the book as a follow up to the various OSX development books I already had. If you have no understanding of Cocoa and want to write a screensaver, this book may not be the place to start.

Given the level of experience required for the book, I thought the first chapter was a waste of time. It is a very high level overview of MacOS in general and MacOS development. Most people already know about Quartz, Project Builder, FreeBSD.

The second chapter at first looked like more of the same, but I found the brief descriptions of the all of the developer applications helpful, if a little shallow. Especially interesting was the listing of command line tools useful to the developer.

Another general thing about the book. Zobkiw is a long time developer of Mac software and this book is plainly written

assuming you are coming from a Mac background. If you're a Windows user or a UNIX guy, some stuff might go right by you, but none of it is really critical.

SECTION 2: APPLICATION LEVEL

The first chapter with code in it is a document-based Cocoa app. This isn't a simple text edit style doc app either, but an app that watches Doppler radar pictures on the web and alerts you when a color in them change. I like that. I don't need another simple simple application, I want something that shows me a lot while it teaches the basics of a document based app.

The book also doesn't go through creation of the project step by step, but assumes you downloaded the project and just need it explained to you. This is appropriate for an advanced audience.

The only problem I had with the app was the document class uses `readFromFile ofType:` and `writeToFile ofType:` instead of `dataRepresentationOf type:` and `loadDataRepresentation of type:` which are what the standard document template creates for you. Nor does he mention these other routines and explain why one would be better or worse than the other.

The next two chapters discuss plug-ins, both Carbon and Cocoa. It is a very useful overview and explanation of how the various plug-in architectures work in OSX.

The last chapter in this section covers Frameworks, showing how to build both a Carbon and Cocoa version of a simple Framework.

SECTION 3: ENHANCING THE SYSTEM

The second section of the book was advanced topics for the application writer. The third section covers how to build unusual non-applications. Specifically System Services, System Preference Panes, Status Items (system-wide menus), Screen Effects (screen savers), and Color pickers.

You've probably noticed the Services sub-menu of a lot of apps. Under 10.1 this was a dead give away you were using a Cocoa, but 10.2 made them available to Carbon apps as well. They are programs, normally faceless, that interact with the current selection. Zobkiw explains how to write one of these services, creating an example that changes the case of the selected text.

Ron Davis is a long time Macintosh Software Engineer, having worked for companies like Apple, and Metrowerks on a variety of products from development tools to anti-virus software. His day job is working for Alsoft, and his evening job is R.A.D. Productions, makers of Suck It Down and FinderEye.

Next Zobkiw talks about how to write a System Preference panel. This allows you to provide a basic UI for faceless application that run all of the time. The example in this chapter sets the preference for the Service created in the previous chapter.

Zobkiw gives an informative discussion of the various kinds of little menus that are displayed in the menu bar on the right side. Some of these are Status Items, some of them Menu Extras. Technically, we developers are only allowed to create Status Items; Menu Extras are reserved for Apple. The example in this chapter is of a Status Item and not a Menu Extra.

I was always a little jealous of how Windows has a standard screen saver mechanism and OS9 didn't. Well that changed in OSX and now you can very easily write those screen savers yourself. They're so easy the chapter explaining them is only 7 pages long. It is still a good introduction that leaves you with a place to start writing your own. If you are going to do a lot of Screen Effects work, you will want to get Saver Lab (<http://www.dozingcat.com/>), which allows you to run your screen saver in a window for debugging.

The last esoteric piece of code in this section is a Color Picker. The OSX Color Picker is actually a collection of pickers. When you open it there is a list of icon buttons at the top of the window that let you pick from a number of pickers. Third party developers can create their own picker panels using Cocoa. Chapter 11 gives you the details on how to do this.

SECTION 4: ADVANCED METHODS

Threads is the first of the advanced topics discussed, which

gives you an overview of what threads are and how they work, and gives you an example of how to use them.

Terminal is the name of the next chapter, but it is really about NSTask and another thread example. Of all the chapters, this was the one I liked the least, because I felt it didn't really cover the topic and was just kind of wrong. He tells you how to use an NSTask to launch a UNIX command line command, but when he does one that you have to wait to return, he uses a thread that polls for the result. It really should have used the notifications NSTask has built into it. That is the right way to do it and the code would have been simpler.

The last two topics covered by their own chapters were XML-RPC and SOAP. These chapters give a basic introduction to using WebServices both in PHP and Cocoa. The PHP for the server made sense, though I was a little surprised Zobkiw did a PHP client. He also builds a client in Cocoa using the WebServices API in CoreServices. These two chapters are very similar. If you have a preference for one remote procedure call methodology over the other, you probably only need to read one of these chapters.

The last chapter of the book is a collection of snippets. Bits and pieces of code that you will find useful, but don't rate a full chapter. This may be the section you find most useful.

SUMMARY

Overall I would say this book is a must have for a complete Cocoa bookshelf. It covers topics no other book covers and does it in a concise, referenceable way.

Who has the best fixed wireless solution in the industry?



There is no comparison!

Effective Data Throughput ⁽¹⁾ <small>(Not to be confused with Signaling Rate)</small>	9.5 Mbps			Dynamic Bandwidth Control ⁽²⁾	Yes
Range ⁽²⁾	Fade margin:	SU	SU-EXT	Latency Control	Yes
	6 dB	7 mi.	20 mi.		
	12 dB	1.5 mi.	10 mi.		
Users/AP	500			RF Thresholding ⁽⁴⁾	Yes
Non-Overlapping Channels	6			User Interface	Telnet, HTTP, Serial
Software Controlled Antenna Polarization	Yes			Price - SU/AP ⁽⁵⁾	\$495/\$895

(1) Measured on SmartBits 600 platform with 1518 byte sized packets.

(2) Trango specifies ranges with a 12 dB fade margin for added robustness.

(3) Indicates the ability to dynamically change up/down stream data throughput for unmatched bandwidth efficiency.

(4) A feature to minimize RF interference not found in competing systems.

(5) Low volume; ask us about higher volume discounts.

Trango Broadband is committed to providing the best technology at the most affordable price for your broadband wireless access deployment. Check us out and see why our Access5800™ solution is the best in class. Give us a call...Don't settle for less anymore!

trangobroadband
WIRELESS
A division of Trango Systems, Inc.

Call Now!

858-653-3900

EMAIL: sales@trangobroadband.com
www.trangobroadband.com

FIXED WIRELESS...THAT WORKS!



(Ask about our third party leasing program)

by Wolfgang Ante, Vienna, Austria.

Using Rendezvous

How to discover and publish services on the LAN

WHAT IS RENDEZVOUS?

To start with a buzzword: Rendezvous is zero-config networking. That means using Rendezvous you should not have to set up any server addresses, user names, and such to access and use services on the local network. Just use them. Apple users not switching over from Windows recently already know this: AppleTalk supported zero-config networking from the beginning. So why is Rendezvous better?

First, Rendezvous is based on IP networking. IP is the industry standard, the Internet protocol, and has far less overhead than most other protocols. On the other hand AppleTalk is a proprietary protocol with a lot of overhead, making it more expensive and slower.

Second, Rendezvous is open. The source code is available for download and you may include it into your software for free. You will not need this as long as your program is running on Mac OS X only. But for developers on Linux and Windows this means they can use Rendezvous, too. If you take a look at the Rendezvous mailing list you can easily see that they do.

PUBLISHING AND BROWSING SERVICES

When your applications wants to provide a service on the network you will have to publish it, when you want to use a service published by someone else you will have to browse for services. A service is whatever you want it to be. Rendezvous is just helping to announce and find services. It gives you all you need to connect, namely the IP address and port number, but doesn't setup the connection itself.

Cocoa gives you a very powerful way to connect to other Cocoa applications on the network called Distributed Objects. My next article will be on Distributed Objects, but for now let's get into the details of using Rendezvous in code.

GOALS

I noticed that Rendezvous is not widely adopted yet, and after working with it myself I think I know why: You still have to care about a lot of things you are not very much interested in, even in Cocoa. Don't get me wrong, I think the Cocoa API to Rendezvous is very good, but for 90% of the applications that could use Rendezvous it would be very handy to have something that gives you just two things: The ability to publish a service and to browse for others who published the same type of service in one easy to use class.

This article will show you how Rendezvous works and additionally will give you a simplified interface (the `ARendezvousController` class) that you are free to use in your software. Using Rendezvous will open your application for teamwork. Think about it, in a lot of applications it makes sense to connect and communicate!

OVERVIEW

Cocoa gives you two classes for the two main objectives: `NSNetService` to publish services and `NSNetServiceBrowser` to browse published services.

Because network operations may take some time to finish, both classes use the concept of delegation to keep your application responsive. You ask for what you want and continue. As soon as the information is available your delegate will be notified about the found information. This also solves the problem of dynamic changes, when services are published or removed later on. Every time the situation changes your delegate will be notified.

There are quite a lot of delegation methods that make understanding the API confusing at first. I will try to concentrate on a typical flow of information, if you are interested in the complete list of delegation methods later, I recommend using the freeware "AppKiDo" from Andy Lee. It is a very good Cocoa API browser, I could not live without it anymore. It is available at <http://homepage.mac.com/aglee/downloads>.

Wolfgang Ante is the founder of ARTIS Software (<http://www.artissoftware.com>). In the last 14 years he worked on Macintosh products for ARTIS and lots of other companies, including a product that won the Macworld's Editors Choice Award. At the time he is trying hard to transform ARTIS into a successful Mac OS X shareware company. Beside that he is always happy to offer his experience for working on your project. You can reach him at wolfgang@artissoftware.com.

ThinkfreeOffice

COMPATIBLE WITH MICROSOFT WORD, EXCEL AND POWERPOINT

WORDPROCESSOR • SPREADSHEET • PRESENTATION GRAPHICS

The Affordable Office Alternative!



Three High-Performance Applications

Thinkfree Write

Thinkfree Write is a powerful word processing application that enables you to create rich, professional quality documents and Web pages. You can insert tables, images, and clipart, or even apply custom layouts to your document...then effortlessly proofread your work with the easy-to-use spelling and auto-correction features.

Thinkfree Calc

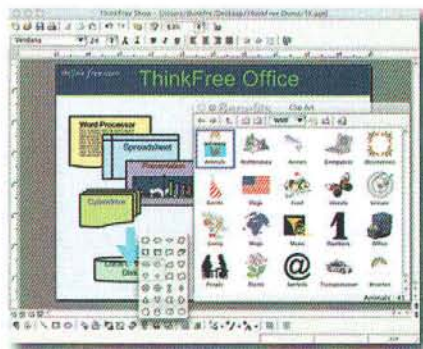
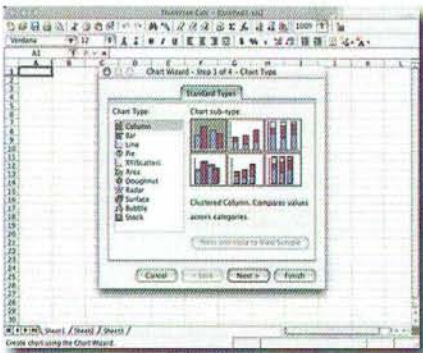
Thinkfree Calc is a full-featured, easy-to-use spreadsheet application that can easily tackle the most complex analytical tasks with over 40 charts and 300 function capabilities. Thinkfree Calc opens, edits, and saves directly into the Microsoft Excel (.xls) format, so users can seamlessly share documents and collaborate with Microsoft Office users.

Thinkfree Show

Thinkfree Show enables you to create high-impact presentations including animation effects, drawings, images, clipart, and other graphic features. Thinkfree Show opens, edits and saves directly into the Microsoft PowerPoint (.ppt) format.

CyberdrivePlus

A free, one-year subscription to CyberdrivePlus is also included. CyberdrivePlus provides you with secure, Internet file storage and free online software upgrades!



"Thinkfree is a best-of-breed program that will exceed your expectations."
— Jeffery Battersby



"Thinkfree Office is the next best thing and then some."
— Deborah Shadovitz



"Thinkfree Office is an impressive attempt to crack the seemingly impenetrable productivity market."
— Chris Ward

amazon.com.

Apple Store



Apple Specialist



ONLY
\$49⁹⁵

PUBLISHING

Publishing a service is quite easy. You create a service by creating an object of type `NSNetService`, set the delegate and publish the service. **Listing 1** shows this:

Listing 1:

Publishing

```
NSNetService *service;
NSString *publishedName;
Int portNumber;

service = [[NSNetService alloc] initWithDomain:@" "
        type:@"_servicetype._tcp."
        name:publishedName
        port:portNumber];
[service setDelegate:delegate];
[service publish];
```

To publish on the default domain (which is the 'local' domain, i.e. your LAN) you pass an empty string to `initWithDomain`. `type` is an identifier for the kind of service you are publishing. The naming convention for this is `"_servicetype._tcp."` where you replace `servicetype` with your identifier. The type is usually not visible to the user, so be as clear or obscure as you like. `name` identifies your service to the network and must be unique. The name usually is visible to the user. Finally `port` must contain a port number you already acquired for the service.

`setDelegate` sets the delegate, this is the object that will be notified later. `publish` will start the process. As soon as you publish your service it will broadcast its presence on the network.

The only interesting delegate method at this point is `netService:didNotPublish:`. This will be sent to your delegate when the service could not be published, otherwise you can assume that your service has been published and is visible to others now.

BROWSING

Browsing for services is a bit more complicated. You start by creating an object of type `NSNetServiceBrowser`, set the delegate and start browsing. **Listing 2** shows this:

Listing 2:

Start browsing

```
NSNetServiceBrowser *browser;

browser = [[NSNetServiceBrowser alloc] init];
[browser setDelegate:self];
[browser searchForServicesOfType:@"_servicetype._tcp."
        inDomain:@" "];
```

First you create the object and set the delegate to be notified, then you start browsing by asking for services of the type you are interested in. As before pass an empty string to browse in the default domain.

Again the requested information will be sent to your delegate later. The interesting ones that will be sent to you at this point are `netServiceBrowser:didFindService:moreComing:` and `netServiceBrowser:didRemoveService:moreComing:`. Rendezvous is dynamic, so services will show up and go away from time to time. As long as you do not stop browsing, your delegate will receive these messages every time something changes about the availability of services of the requested type on the network.

Lets start by looking at `netServiceBrowser:didFindService:moreComing:`. **Listing 3** shows an implementation:

Listing 3:

Service found

```
-(void)netServiceBrowser:
(NSNetServiceBrowser *)aNetServiceBrowser
didFindService:(NSNetService *)aNetService
moreComing:(BOOL)moreComing
{
    [aNetService setDelegate:self];
    [aNetService resolve];
}
```

You receive three parameters: `aNetServiceBrowser` is the browser that sent the message, `aNetService` is the service that has been found and `moreComing` tells you if there are more queued service objects. The `moreComing` parameter looks a bit strange at first, but can be used as a hint to your user interface code to delay updating.

The information you are probably most interested in comes as the second parameter and it looks like this is already everything you need. Unfortunately you only get one third of the information you are interested in. You can send the service object a `name` message to get the name of the service. To obtain the IP address and port number you have to set a delegate and

send a **resolve** message to the service. Don't forget to set the delegate, since this time the service will sent you messages, not the browser. When you send **resolve** this will again result in messages sent to the delegate. The interesting one this time is **netServiceDidResolveAddress:**.

Following the successful path lets start by looking at **netServiceDidResolveAddress:**. **Listing 4** shows an implementation:

Listing 4:

```

                                Addresses resolved
- (void)netServiceDidResolveAddress:(NSNetService *)sender
{
    NSString      *name = nil;
    NSData        *address = nil;
    struct sockaddr_in *socketAddress = nil;
    NSString      *ipString = nil;
    int            port;

    for (i = 0; i < [[sender addresses] count]; i++)
    {
        name = [sender name];
        address = [[sender addresses] objectAtIndex:i];
        socketAddress = (struct sockaddr_in *)
            [address bytes];
        ipString = [NSString stringWithFormat:@"%s",
            inet_ntoa(socketAddress->sin_addr)];
        port = socketAddress->sin_port;
        [...]
    }
}

```

The only parameter is the service object that sent the message. Now the service object holds all the needed information to connect to the service. You obtain the name by

sending a **name** message to the service. To find the IP address and port number you send an **addresses** message to the service. You receive an array containing all addresses. This array usually contains exactly one address, but to be on the save side loop through all addresses in the array. An address is of type **sockaddr_in**. This is a standard networking structure of Unix. See the above code for the details. After you have the name, IP and port you are ready to use that information. You will probably store the information in an array so you can access it any time you need it.

Lets go one step back to **netServiceBrowser:didRemoveService:moreComing:**. This message will be sent to you every time a service has been removed. If you stored the information in an array you have to remove the entry for that service to keep the array up-to-date.

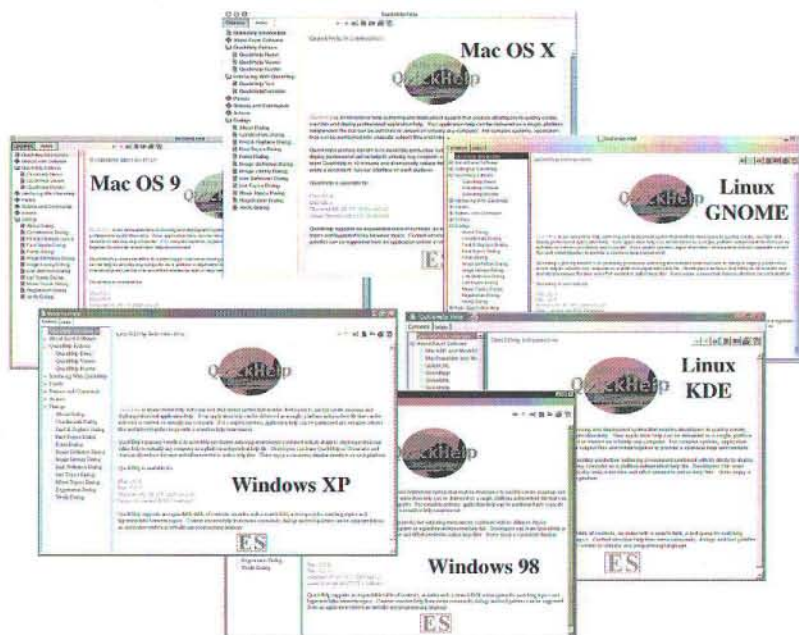
BROWSING PROBLEMS

On Mac OS X 10.2.3 and before there is a problem with service browsing and switching locations. When you switch the location from the Apple menu (or change your IP address in any other way) the browser gets seriously confused. Your old and new IP may or may not show up. You may get entries that are simply wrong, pointing to your old IP address.

This happens with Apple's Rendezvous-enabled products, too. I saw myself showing up in iChat double after switching back and forth and Apple's AFPServer (the 'Personal File Sharing' server) also has problems with it, when you look at your logs.

QuickHelp™

**Author Anywhere
Deploy Everywhere**

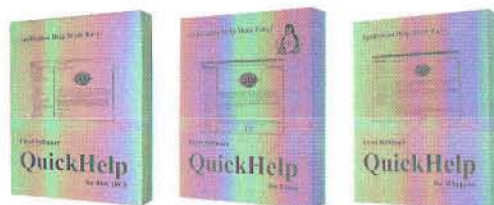


Help Authoring:

- Author and Test Help From One Window
- Add Color Highlighted Text and Images
- Hyperlink Words to Other Topics
- Associate Index Words With Topics
- Check For Missing Links or Images
- Import or Export Topics as a Text File
- Make Sections or Entire Topics Conditional

Help Viewing:

- Expand, Collapse and Navigate Contents
- Click Back, Forward or Hyperlink To Topics
- Locate Topics By Index or Word Search
- Get Context Sensitive Application Help
- Enjoy Familiar Interface on Every Platform



Excel Software
www.excelsoftware.com
 Ph. 505-771-3719

It might be that this is already fixed when you read this, since Apple has released fixes to Rendezvous with every of the last three system updates. I read it is much better with Mac OS X 10.2.5, but to be nice to your users you should try to offer them a workaround on systems before 10.2.5 at least.

My suggestion is to offer refresh functionality. A refresh starts a complete new search forgetting about the cached entries.

If you want to automate this, you could regularly check if your IP address has changed (use `[[NSHost currentHost] address]` for this) and then refresh. The `ARendezvousController` class does not do this, but it offers a `refreshBrowsing` method that you can hook up to a button in your user interface easily.

MAKING THINGS EASIER

By now you have a good overview of how to use Rendezvous. But like me, I guess you will think that you don't want to weave that all into your code every time you want to use Rendezvous.

Therefore I came up with `ARendezvousController`. It has a simplified interface and still provides everything to publish and browse services. An `ARendezvousController` object cares about one service type, publishing and browsing it. It provides start/stop methods for publishing and browsing. It has exactly one delegate method that is called every time something changes. It keeps track of the found services and on request returns an array with all found services, ready with name, IP address and port number. It even creates the socket others will connect to for you.

Listing 5:

Public interface of `ARendezvousController`

```
// public interface
@interface ARendezvousController : NSObject

- (id)initWithName:(NSString *)name type:(NSString *)type
  port:(int)port;
- (void)dealloc;

- (id)delegate;
- (void)setDelegate:(id)object;
- (NSString *)name;
- (BOOL)setName:(NSString *)name;
- (NSSocketPort *)socket;

- (void)activateBrowsing:(BOOL)flag;
- (BOOL)isBrowsing;
- (void)refreshBrowsing;
- (void)activatePublishing:(BOOL)flag;
- (BOOL)isPublished;
- (NSArray *)discoveredServicesWithInfo;

- (NSString *)ipForName:(NSString *)name;
- (int)portForName:(NSString *)name;

@end

// to be implemented by the delegate
@interface NSObject (RendezvousControllerDelegate)

- (void)discoveredServicesDidChange:(id)sender;

@end
```

USING ARENDEZVOUSCONTROLLER

Instead of going through the list method by method, let's start with a real world example. The following code will fill an `NSTableView` with all available services of type `@ "_demo_tcp."` on your local network. The table view will be updated dynamically every time services are published and removed without user intervention. Just like you know it from iChat.

The complete demo project including `ARendezvousController` can be downloaded from <http://www.artissoftware.com/rendezvous>. This is probably the best way to proceed, since you don't have to create the .nib file in Interface Builder yourself.

If you are reading this without access to the Internet, here is what you need to create in Interface Builder: The main window contains an `NSTableView`. The `NSTableView` has three columns. The identifiers of the three columns should be "name", "ip" and "port".

Listing 6:

`ADemoController.h`

```
@interface ADemoController : NSObject
{
    IBOutlet NSWindow      *_mainWindow;
    IBOutlet NSTableView    *_tableView;

    ARendezvousController *_rendezvousController;
    NSArray                *_lastState;
}

- (void)awakeFromNib;
- (void)discoveredServicesDidChange:(id)sender;
- (int)numberOfRowsInTableView:(NSTableView *)tableView;
- (id)tableView:(NSTableView *)tableView
  objectValueForTableColumn:(NSTableColumn *)tableColumn
  row:(int)row;

@end
```

In Interface Builder import this header file and connect the `_mainWindow` outlet to your window and the `_tableView` outlet to your table view.

Let's start by looking at `awakeFromNib`. This method contains all the necessary setup:

Listing 7:

`awakeFromNib`

```
- (void)awakeFromNib
{
    _lastState = [[NSArray alloc] init];

    _rendezvousController = [[ARendezvousController alloc]
      initWithName:NSFullUserName() type:@"_demo_tcp."
      port:12345];
    [_rendezvousController setDelegate:self];
    [_rendezvousController activateBrowsing:YES];
    [_rendezvousController activatePublishing:YES];

    [_mainWindow center];
    [_mainWindow makeKeyAndOrderFront:self];
}
```

`_lastState` will contain a copy of the last state returned by the `rendezvousController`. In the beginning there are no services, so it is initialized as an empty array. After that the `ARendezvousController` is initialized with a type of

Multiple formats. Multiple platforms. Complex installers.

Aladdin solves the compression and installation puzzle.

**Trying to figure out how to handle multiple
compression formats and platforms?**

The Stuffit Engine solves the compression puzzle.



Aladdin's Stuffit Engine SDK:

- Adds value to your application by integrating powerful compression and encryption.
- Is the only tool that supports the Stuffit file format.
- Provides a single API that supports over 20 compression and encoding formats common on Macintosh, Windows, and Unix.
- Makes self-extracting archives for either Macintosh or Windows.
- Available for Macintosh, Windows, Linux, or Solaris.

Licenses start as low
as \$99/year

To learn more, visit:
www.stuffit.com/sdk/

Stuffit Engine SDK™ The power of Stuffit in your software.



**Looking for the easiest and fastest
way to build an installer?**

Stuffit InstallerMaker completes your puzzle.

It's not enough just to write solid code anymore. You still have to write an installer for your users. Stuffit InstallerMaker makes it simple and effective.

- Stuffit InstallerMaker gives you all the tools you need to install, uninstall, resource-compress or update your software in one complete, easy-to-use package.
- Add marketing muscle to your installers by customizing your electronic registration form to include surveys and special offers.
- Make demoware in minutes. Create Macintosh OS X and Macintosh Classic compatible installers with Stuffit InstallerMaker.

Prices start at \$250

To learn more, visit:
[www.stuffit.com/
installermaker/](http://www.stuffit.com/installermaker/)

Stuffit InstallerMaker™ The complete installation solution.™



www.stuffit.com
(831) 761-6200

© 2002 Aladdin Systems, Inc. Stuffit, Stuffit InstallerMaker, and Stuffit Engine SDK are trademarks of Aladdin Systems, Inc. The Aladdin logo is a registered trademark. All other products are trademarks or registered trademarks of their respective holders. All Rights Reserved.

@ "_demo_tcp." and with port 12345. The port number is just a suggestion. If this port is already in use the number is increased as long as a socket can be created. This is no problem, since the port number is published. At this time the socket will be created. To obtain the socket you would send a `socket` message to the `rendezvousController`. Finally the delegate is set to `self` so we are notified of changes, browsing is started by sending `activateBrowsing:` and the service is published by sending `activatePublishing:` to the `rendezvousController`.

The next method is the delegate method.

Listing 8:

The delegate method

```
(void)discoveredServicesDidChange:(id)sender
{
    [_lastState autorelease];
    _lastState = [[rendezvousController
        discoveredServicesWithInfo] retain];
    [_tableView reloadData];
}
```

It simply releases the old copy of states, asks the `rendezvousController` for an up-to-date array of states and retains that. After that it tells the table view to reload its data.

Finally these two methods fill the table view with data:

Listing 9:

TableView methods

```
(int)numberOfRowsInTableView:(NSTableView *)tableView
{
    return [_lastState count];
}
```

Presto Vivace, Inc.

Fast and Lively Public Relations

Presto Vivace specializes in public relations for small technology companies. Our press contacts database is now available for companies to manage their own publicity.

For only \$99, you can use our professional database to place your press releases. Available in AppleWorks format; e-mail marshall@prestovivace.biz for sample.

4902 Powell Road, Fairfax, VA 22032

703/426-5876, fax 426-5892

<http://www.prestovivace.biz/>

```
(id)tableView:(NSTableView *)tableView
objectValueForTableColumn:(NSTableColumn *)tableColumn
row:(int)row
{
    return [[_lastState objectAtIndex:row]
        valueForKey:[tableColumn identifier]];
}
```

The last one is a wonderful demonstration of applied Cocoa magic, but to understand it you should know first that the array returned by `discoveredServicesWithInfo` contains an `NSDictionary` for every service. Such a dictionary contains three key/value pairs: Key "name" contains an `NSString` with the name of the service. Key "ip" contains an `NSString` with the IP address (like @ "192.168.0.1")

Key "port" contains an `NSNumber` with the port number

By setting the column identifiers of the table view to these keys the data can be fed to the table view with this one-liner.

That's it! It is that simple. In my next article I will show how to send messages over the network just like you would send it to your objects. This is called Distributed Objects. Stay tuned!

Listing 10:

ARendezvousController.h

```
#import <Foundation/Foundation.h>

// interface of the rendezvous class
@interface ARendezvousController : NSObject
{
    NSNetService *_service;
    NSString *_serviceName;
    NSString *_serviceType;
    BOOL _browsing;
    BOOL _publishing;

    NSNetServiceBrowser *_serviceBrowser;
    NSNetServiceBrowser *_domainBrowser;

    NSSocketPort *_socketPort;
    int _portNumber;

    NSMutableArray *_discoveredServicesWithInfo;

    id _delegate;
}

- (id)initWithName:(NSString *)name type:(NSString *)type
port:(int)port;
- (void)dealloc;

- (id)delegate;
- (void)setDelegate:(id)object;
- (NSString *)name;
- (BOOL)setName:(NSString *)name;
- (NSSocketPort *)socket;

- (void)activateBrowsing:(BOOL)flag;
- (BOOL)isBrowsing;
- (void)refreshBrowsing;
- (void)activatePublishing:(BOOL)flag;
- (BOOL)isPublished;
- (NSArray *)discoveredServicesWithInfo;

- (NSString *)ipForName:(NSString *)name;
- (int)portForName:(NSString *)name;

@end

// to be implemented by the delegate
@interface NSObject (RendezvousControllerDelegate)

- (void)discoveredServicesDidChange:(id)sender;

@end
```

Listing 11:

```

A Rendezvous Controller.m

#import "ARendezvousController.h"

#include <netinet/in.h>
#include <arpa/inet.h>

@interface ARendezvousController
(ARendezvousControllerInternal)

// creation
- (void)createSocket;
- (void)createBrowser;
- (void)createService;

// management
- (BOOL)addInfoService:(NSNetService *)
  service name:(NSString *)name ip:(NSString *)ip
  port:(int)port;
- (BOOL)removeInfoService:(NSNetService *)service;

// other
- (void)startBrowsing;

@end

@interface ARendezvousController (NSNetServiceDelegation)

// publication
- (void)netService:(NSNetService *)sender
  didNotPublish:(NSDictionary *)errorDict;
- (void)netServiceWillPublish:(NSNetService *)sender;
- (void)netServiceDidStop:(NSNetService *)sender;

// resolution
- (void)netService:(NSNetService *)sender
  didNotResolve:(NSDictionary *)errorDict;
- (void)netServiceDidResolveAddress:(NSNetService *)sender;
- (void)netServiceWillResolve:(NSNetService *)sender;

```

```

@end

@interface ARendezvousController
(NSNetServiceBrowserDelegation)

// browsing
- (void)netServiceBrowser:(NSNetServiceBrowser *)
  aNetServiceBrowser didFindService:(NSNetService *)
  aNetService moreComing:(BOOL)moreComing;
- (void)netServiceBrowser:(NSNetServiceBrowser *)
  aNetServiceBrowser didNotSearch:(NSDictionary *)errorDict;
- (void)netServiceBrowser:(NSNetServiceBrowser *)
  aNetServiceBrowser didRemoveService:(NSNetService *)
  aNetService moreComing:(BOOL)moreComing;
- (void)netServiceBrowser:(NSNetServiceBrowser *)
  aNetServiceBrowser didFindDomain:(NSString *)domainString
  moreComing:(BOOL)moreComing;
- (void)netServiceBrowser:(NSNetServiceBrowser *)
  aNetServiceBrowser didRemoveDomain:(NSString *)domainString
  moreComing:(BOOL)moreComing;
- (void)netServiceBrowserDidStopSearch:
  (NSNetServiceBrowser *)aNetServiceBrowser;
- (void)netServiceBrowserWillSearch:
  (NSNetServiceBrowser *)aNetServiceBrowser;

@end

@implementation ARendezvousController

- (id)initWithName:(NSString *)name type:(NSString *)type
  port:(int)port
{
    self = [super init];
    if (self)
    {
        // store name, type and port for later
        [self setName:name];
        _serviceType = [type retain];
        _portNumber = port;
    }
}

```

ListSTAR®

www.liststar.com

The most flexible email processing system available. Easily create mailing lists or email-on-demand services. Use built in rules and/or AppleScript/AppleEvents to handle any email task, no matter how simple or complex. Demo available.

MacRADIUS™

www.macradius.com

The easiest to use RADIUS server available. The groups feature allows you to make changes for a large number of users in one easy step. Enabling or disabling access for a user or a group of users is a one click operation, without having to stop the server. Support for AppleScript/AppleEvents makes it easy to control MacRADIUS. Demo available.

SimpleText Filter

Plug-in for EIMS*

www.mcfsoftware.com/stf/

Easy to use header and content filtering. Scan incoming messages for sequences of text, digits, etc. Base64 messages are decoded so you can check for content despite attempts to hide the text. Demo available.

Auto Reply

Plug-in for EIMS*

www.mcfsoftware.com/ar/

This plug-in allows you to easily set up auto-reply messages for users. Addresses that have been sent auto-reply messages are tracked, preventing auto-reply message loops. Demo available.

**Address List
Sorter**

www.mcfsoftware.com/als/

Fast and powerful utility for sorting and cleaning email lists. Sort email address lists in alphabetical or domain order, remove duplicates and improperly formed addresses. Demo available.

*EIMS—Eudora Internet Mail Server (www.eudora.co.nz)



...*simply
dependably
engineered*

www.mcfsoftware.com

```

// initialize array
_discoveredServicesWithInfo =
    [[NSMutableArray alloc] init];

// create the socket and the browser
[self createSocket];
[self createBrowser];
}
return self;
}

- (void)dealloc
{
    // clean up
    [self activatePublishing:NO];
    [_serviceBrowser stop];
    [_domainBrowser stop];
    [_socketPort release];
    [_discoveredServicesWithInfo release];
    [super dealloc];
}

- (id)delegate
{
    return _delegate;
}

- (void)setDelegate:(id)object
{
    // find if delegate supports the delegation message
    if (![object respondsToSelector:
        @selector(discoveredServicesDidChange:)])
        NSLog(@"Delegate does not respond to
            'discoveredServicesDidChange:'");

    _delegate = object;
}

- (NSString *)name
{
    return _serviceName;
}

- (BOOL)setName:(NSString *)name
{
    // change name only when not already published
    if (!_publishing)
    {
        [name retain];
        [_serviceName release];
        _serviceName = name;
        return YES;
    }
    else
    {
        NSLog(@"Cannot change name while service
            is published!");
        return NO;
    }
}

- (NSSocketPort *)socket
{
    return _socketPort;
}

- (void)activateBrowsing:(BOOL)flag
{
    // if requested and actual state match don't proceed
    if (flag == _browsing)
        return;

    if (flag)
    {
        // activate browsing
        _browsing = YES;
        [_serviceBrowser searchForServicesOfType:
            _serviceType inDomain:@"*"];
    }
    else
    {
        // deactivate browsing
        _browsing = NO;
        [_serviceBrowser stop];
        [_discoveredServicesWithInfo removeAllObjects];
        [_delegate discoveredServicesDidChange:self];
    }
}

- (BOOL)isBrowsing
{
    return _browsing;
}

- (void)startBrowsing
{
    // should only be called when browsing off
    if (!_browsing)
    {
        NSLog(@"Browsing already started!");
        return;
    }

    // start browsing
    _browsing = YES;
    [_serviceBrowser searchForServicesOfType:
        _serviceType inDomain:@"*"];
}

- (void)refreshBrowsing
{
    // don't refresh if not browsing
    if (!_browsing)
        return;

    // start/stop
    // (stop is deferred to the end of message queue)
    [self activateBrowsing:NO];
    [self performSelector:@selector(startBrowsing)
        withObject:nil afterDelay:0.0];
}

- (void)activatePublishing:(BOOL)flag
{
    // if already activated then don't do anything
    if (_publishing == flag)
        return;

    if (flag)
    {
        // activate service
        [self createService];
    }
    else
    {
        // deactivate service
        [_service stop];
        _service = nil;
    }

    // set new state
    _publishing = flag;
}

- (BOOL)isPublished
{
    return _publishing;
}

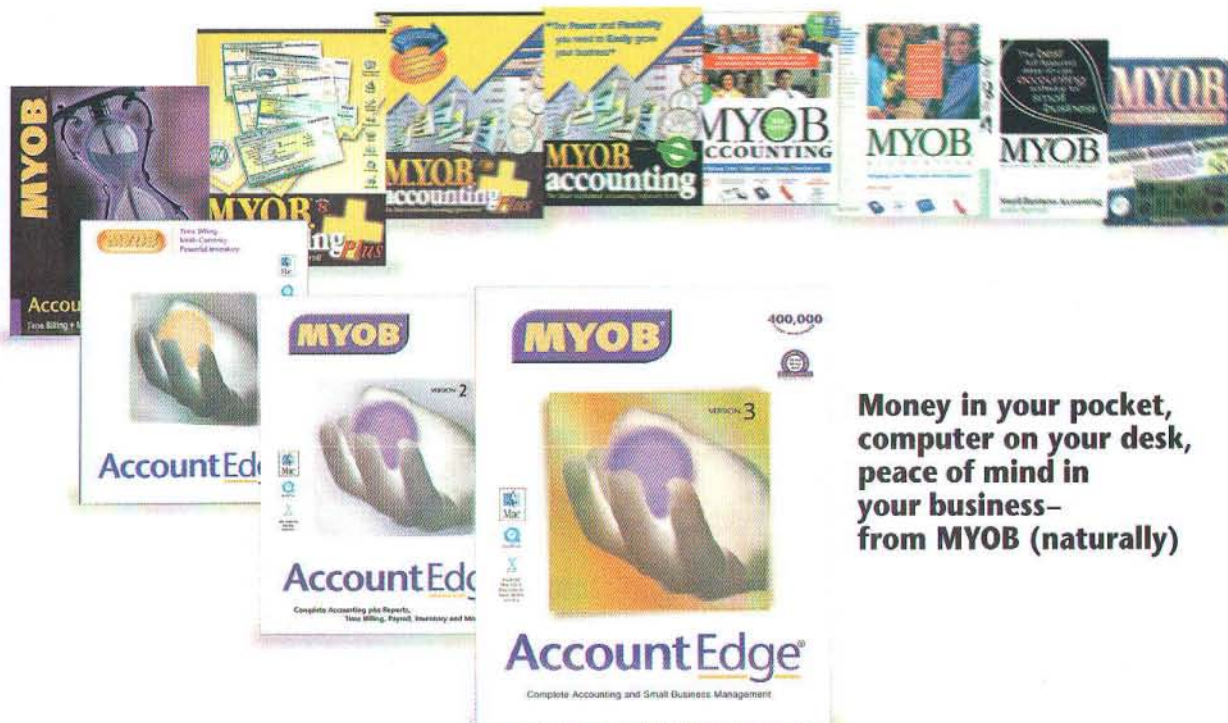
- (NSArray *)discoveredServicesWithInfo
{
    // return (autoreleased) copy
    return [NSArray
        arrayWithArray:_discoveredServicesWithInfo];
}

- (NSString *)ipForName:(NSString *)name
{
    NSEnumerator *e = nil;
    NSDictionary *dict = nil;

    // find the corresponding ip to the given name
    e = [_discoveredServicesWithInfo objectEnumerator];
    while (dict = [e nextObject])
}

```

\$500 Cash Back!



**Money in your pocket,
computer on your desk,
peace of mind in
your business—
from MYOB (naturally)**

Evolving with the Mac since 1989

Hurry! Offer valid May 1 - June 30, 2003

Get \$249 back!

When you buy AccountEdge 3 and up to \$249 (or more) of other merchandise of your choosing from your reseller on the same invoice. Claim up to \$249 cash back from MYOB US, Inc. when you enroll in one year of MYOB All Inclusive Coverage on a convenient monthly billing plan.

For claim form, complete details and eligibility rules, visit **www.myob.com/us** Or call **800-322-6962**

Get \$500* back!

When you buy AccountEdge 3 and any Mac computer on the same invoice.

*** With two year enrollment in MYOB All Inclusive Coverage on a convenient monthly billing plan.**

For claim form, complete details and eligibility rules, visit **www.myob.com/us** Or call **800-322-6962**

www.myob.com/us

MYOB

800-322-MYOB (6962)

Small Business Management and Accounting

Offer details available online at www.myob.com/us. Valid for AccountEdge retail product #meurm purchased between May 1 and June 30, 2003. Offer valid only with purchase of MYOB All Inclusive Coverage plan as stated in the claim form. Cash back payment schedule as affixed in claim details. Cannot be combined with any other offer. Void where prohibited. MYOB, the MYOB logo and AccountEdge are registered trademarks of MYOB Ltd. ©MYOB 2003

```

        if ([[dict objectForKey:@"name"]
            isEqualToString:name])
            return [dict objectForKey:@"ip"];

// return nil when not found
        return nil;
    }

- (int)portForName:(NSString *)name
{
    NSEnumerator *e = nil;
    NSDictionary *dict = nil;

    // find the corresponding ip to the given name
    e = [_discoveredServicesWithInfo objectEnumerator];
    while (dict = [e nextObject])
        if ([[dict objectForKey:@"name"]
            isEqualToString:name])
            return [[dict objectForKey:@"port"] intValue];

// return 0 when not found
    return 0;
}

@end

@implementation ARendezvousController
(RendezvousControllerInternal)

- (void)createSocket
{
    // already there
    if (_socketPort)
        return;

    // look for free port
    while (!_socketPort)
    {
        _socketPort = [[NSSocketPort alloc]
            initWithTCPPort:_portNumber];
        _portNumber++;
    }
    _portNumber--;
}

- (void)createService
{
    // already there
    if (_service)
        return;

    // create service, make self the delegate and publish
    _service = [[NSNetService alloc] initWithDomain:@""
        type:_serviceType name:_serviceName port:_portNumber];
    [_service setDelegate:self];
    [_service publish];
}

- (void)createBrowser
{
    // setup service browser
    if (!_serviceBrowser)
    {
        _serviceBrowser = [[NSNetServiceBrowser alloc]
            init];
        [_serviceBrowser setDelegate:self];
    }
}

- (BOOL)addInfoService:(NSNetService *)service name:(NSString *)name ip:(NSString *)ip port:(int)port
{
    NSMutableDictionary *dict = nil;
    NSEnumerator *e = nil;

    // if already there then don't add
    e = [_discoveredServicesWithInfo objectEnumerator];
    while (dict = [e nextObject])
        if ([[dict objectForKey:@"ip"] isEqualToString:ip]
            if ([[dict objectForKey:@"port"] intValue] ==
                port)
                return NO;
}

```

```

// add if not found in array
dict = [NSMutableDictionary dictionary];
[dict setObject:ip forKey:@"ip"];
[dict setObject:[NSNumber numberWithInt:port]
    forKey:@"port"];
[dict setObject:name forKey:@"name"];
[dict setObject:service forKey:@"service"];
[_discoveredServicesWithInfo addObject:dict];
return YES;
}

- (BOOL)removeInfoService:(NSNetService *)service
{
    NSDictionary *dict = nil;
    NSEnumerator *e = nil;
    NSMutableArray *delete = nil;

    // look for object with this service and mark for delete
    delete = [NSMutableArray array];
    e = [_discoveredServicesWithInfo objectEnumerator];
    while (dict = [e nextObject])
        if ([[dict objectForKey:@"service"]
            isEqualToString:service])
            [delete addObject:dict];

    // delete marked objects
    e = [delete objectEnumerator];
    while (dict = [e nextObject])
        [_discoveredServicesWithInfo removeObject:dict];

    return NO;
}

@end

@implementation ARendezvousController
(NSNetServiceDelegation)

- (void)netService:(NSNetService *)sender
    didNotPublish:(NSDictionary *)errorDict
{
    // publishing failed
    _publishing = NO;
    NSLog(@"Publishing the service %@ failed.",
        [sender name]);
    [_delegate discoveredServicesDidChange:self];
}

- (void)netServiceWillPublish:(NSNetService *)sender
{
    // does nothing for now,
    // implemented for your possible additions
}

- (void)netServiceDidStop:(NSNetService *)sender
{
    // does nothing for now,
    // implemented for your possible additions
}

- (void)netService:(NSNetService *)sender
    didNotResolve:(NSDictionary *)errorDict
{
    // resolving failed
    NSLog(@"Resolving of address for service %@ failed.",
        [sender name]);
}

- (void)netServiceDidResolveAddress:(NSNetService *)sender
{
    NSData *address = nil;
    struct sockaddr_in *socketAddress = nil;
    NSString *ipString = nil;
    int port, i;

    for (i = 0; i < [[sender addresses] count]; i++)
    {
        // gather data about this published service
        address = [[sender addresses] objectAtIndex:i];
        socketAddress = (struct sockaddr_in *)
            [address bytes];
    }
}

```

```

    ipString = [NSString stringWithFormat:@"%s",
        inet_ntoa (socketAddress->sin_addr)];
    port = socketAddress->sin_port;

    // published localhost is a Rendezvous strangeness:
    // ignore that!
    if ([ipString isEqualToString:@"127.0.0.1"])
        continue;

    // notify delegate of change
    if ([self addInfoService:sender name:[sender name]
        ip:ipString port:port])
        [_delegate discoveredServicesDidChange:self];
}

- (void)netServiceWillResolve:(NSNetService *)sender
{
    // does nothing for now,
    // implemented for your possible additions
}

@end

@implementation ARendezvousController
(NSNetServiceBrowserDelegation)

- (void)netServiceBrowser:(NSNetServiceBrowser *)
aNetServiceBrowser didFindService:(NSNetService *)
aNetService moreComing:(BOOL)moreComing
{
    // add to discovered services and resolve it
    [aNetService setDelegate:self];
    [aNetService resolve];
}

- (void)netServiceBrowser:(NSNetServiceBrowser *)
aNetServiceBrowser didRemoveService:(NSNetService *)
aNetService moreComing:(BOOL)moreComing
{
    // remove and notify
    [self removeInfoService:aNetService];
    [_delegate discoveredServicesDidChange:self];
}

- (void)netServiceBrowserDidStopSearch:
(NSNetServiceBrowser *)aNetServiceBrowser
{
    // empty the arrays and notify delegate
    if (aNetServiceBrowser == _serviceBrowser)
    {

```

```

        [_discoveredServicesWithInfo removeAllObjects];
        [_delegate discoveredServicesDidChange:self];
    }
}

- (void)netServiceBrowser:(NSNetServiceBrowser *)
aNetServiceBrowser didNotSearch:(NSDictionary *)errorDict
{
    NSLog(@"Unable to search.");
}

- (void)netServiceBrowserWillSearch:(NSNetServiceBrowser *)
aNetServiceBrowser
{
    // does nothing for now,
    // implemented for your possible additions
}

- (void)netServiceBrowser:(NSNetServiceBrowser *)
aNetServiceBrowser didFindDomain:(NSString *)domainString
moreComing:(BOOL)moreComing
{
    // does nothing for now,
    // implemented for your possible additions
}

- (void)netServiceBrowser:(NSNetServiceBrowser *)
aNetServiceBrowser didRemoveDomain:(NSString *)domainString
moreComing:(BOOL)moreComing
{
    // does nothing for now,
    // implemented for your possible additions
}

@end

```

BIBLIOGRAPHY AND REFERENCES

Apple Computer. "Rendezvous Network Services"
<http://developer.apple.com/techpubs/macosx/Cocoa/TasksAndConcepts/ProgrammingTopics/NetServices/index.html>

Michael Beam. "Incorporating Rendezvous into Your Cocoa Applications"

<http://www.macdevcenter.com/pub/a/mac/2002/11/08/cocoa.html>



Valentina

Object-Relational SQL Database

The fastest database engine for MacOS/Windows

**It operates 100's and sometimes
a 1000 times faster than other systems**

www.paradigmasoft.com
 Hosted by macserve.net
 Download full featured evaluation version

Get more out of your Macintosh!

DevDepot sells the tools, toys and technology to put more muscle into your Mac. Innovative upgrades, essential accessories, and cutting edge new products for those of you who just want more.

Visit the Official Store, managed by DevDepot, for show specials and great new products from all over the show floor!

DevDepot has it all!

More Storage



500 GB on your Desktop

The BigDrive from LaCie, FireWire 800 and all the capacity you can ask for.

Put 80 GB in your PowerBook

Why leave it behind? Keep all the data and applications you need on your PowerBook or iBook.



Add a SuperDrive

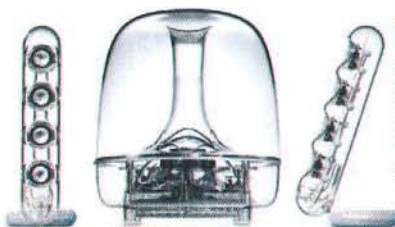
An internal SuperDrive does it all – DVD-R/RW and CD-R/RW in one drive. iDVD compatible, easily to install.

Lucid FireWire Powered CD-R/DVD-R

Lucid CD-R/RW and DVD-R/RW drives are iTunes compatible and bus powered. Portable and Powerful!



Better Sound



Harman Kardon USB Speakers

No audio out, no problem! The best looking, great sounding speakers for USB Macs.

JBL Invader Speakers

Dolby 4.1 speakers and amp for your Mac or iPod!



PowerWave USB Amp

Audio interface and amplifier for your Mac or PC, connect your home speakers or sound system to your USB Mac.

Noise Reduction Headphones

With airline adapter, the perfect companion for iPod and traveling with your PowerBook.



More Speed



HARMONi for iMac

Pump your original iMac to 600 MHz G4 and add FireWire!

G4 Upgrades

Put up to a 1.2 GHz G4 in your box. Upgrades for your G4 Tower, your iMac, Blue & White, Beige G3, 8500 and before!



Macintosh RAM

RAM upgrades at great prices, designed and built for best performance and 100% compatibility with your Macintosh.



Wallstreet G4

Your Wallstreet G3 PowerBook can have a 500 MHz G4 Processor!



New Interfaces

PowerMate Analog Control

Control iTunes or DVD Player, edit home movies or just use it next to your mouse as the 'ultimate' scroll wheel!



Wireless Mice

Finally a wireless mouse for Mac, done right! Portable version for PowerBook users or desktop version!



AirPort USB Adapter

Add AirPort connectivity to your iMac, PowerMac or any USB Macintosh!



iceCAD

Ideal tool for Apple InkWell for handwriting recognition, freehand sketches, or use with graphic applications.



PowerBook & iPod Innovations

Replacement Batteries

Long flight or multiple meetings, do you need more juice? Batteries for G3 (Pismo, Lombard, Wallstreet) and G4 PowerBooks!



The PowerBook Handle

A must have for G4 PowerBook users, put a handle on your PowerBook and great to prop up your PowerBook for a comfortable tilt.



iTrip

The ultimate iPod accessory, plugs into your iPods FireWire and Audio ports. FM broadcaster, powered and controlled through your iPod.



Better AC Adapters

The smallest, most compact AC adapter for PowerBooks and iBooks. One adapter for home, one for the office!



DEV DEPOT®

www.devdepot.com

877-DEPOT-NOW

By Andrew S. Downs

Revolution OS

The rise of the free software and open source movements

INTRODUCTION

Every so often a movie comes along that tells a techie tale in just the right way. Making the rounds right now at film festivals and on various movie channels is an independent film called *Revolution OS*. Produced and directed by J.T.S. Moore, *Revolution OS* tells the story of the rise of the Free Software and Open Source movements. Using a combination of news bites, camera shots of Silicon Valley, and occasional statistics illustrating the increasing adoption of Linux and popularity of Linux-related Initial Public Offerings, the movie intersperses such teasers with interviews with key players in these movements.

The movie opens with Eric Raymond recounting an episode in which he encountered the Microsoft VP of Consumer Products at a conference. Eric concludes the encounter by telling the VP "I'm your worst nightmare." This opening exchange sets the tone for the movie: a growing movement that threatens the dominance of Microsoft.

The talking head format used for these interviews works better than you might expect. The main players are introduced with little fanfare. This allows you to focus on the message that each relays. We meet several of the players in the opening minutes: Eric Raymond, Linus Torvalds, Bruce Perens, and Richard Stallman. The interviews show the passion as well as the facts. The small amount of narration and voiceover provides some variety in the presentation, and in no way detracts from the message. There is no evidence of "Hollywood" in this film.

The tech talk is light. The most complex topic discussed is the distinction between the monolithic kernel architecture of Linux and the microkernel-based GNU HURD. If you read Cliff Stoll's *The Cuckoo's Egg*, the superficial treatment of potentially complex topics is similar though engaging. This makes the movie accessible to a broader audience.

For example, Bruce Perens, the author of the Open Source Definition, discusses open source as a way for developers to collaborate on projects without restrictive intellectual property laws and contracts getting in the way. Those developers sacrifice intellectual property rights in an effort to increase the

number of users of the software. This could easily have gotten bogged down in legal terminology, but Bruce makes the topic easily accessible.

We also find out that the connections between the philosopher Stallman, the engineer Torvalds, and the companies that aim to bring these products to market are based not on a bandwagon mentality but on real experience with the underlying products. Michael Tiemann of Cygnus Software and Larry Augustin of VA Linux Systems, who serve as the film's primary entrepreneurs, were both programmers who contributed to the GNU software code base earlier in their careers. No doubt this enhanced their ability to make otherwise free tools a commercial success.

Another indication of the symbiosis between companies and philosophers is Netscape's decision to release its browser source code (the Mozilla project) as open source based on the principles espoused in Eric Raymond's *The Cathedral and the Bazaar*. Fearing a Microsoft monopoly and perversion of HTML and HTTP, Netscape became the first commercial vendor to offer its source code as a product.

FREE VS. OPEN SOURCE

The movie goes to great lengths to balance the discussion between the Free Software Movement and Open Source. Although Linux and Open Source receive the lion's share of media attention, we find out that Richard Stallman and the Free Software Foundation predate Open Source by ten years or so.

Free software and open source are not synonymous. Free software is more of a political stance than an economic one: "free" refers not to price but to philosophy. Free software can be used and modified without restriction. In the proprietary, single vendor or non-free software world we know this concept as piracy, except where very liberal licenses apply. Free software began as Richard Stallman's personal crusade about twenty years ago. Interviews with Richard in the movie clearly bring out the zeal and passion in the man. What started as a configurable text editor (Emacs) in the 1980s evolved along several non-contiguous paths, including the Free Software Foundation's GNU project and the Open Source movement where Linux and Apache remain the most obvious examples of successful projects into an essential element of our technology

Andrew writes embedded system software in New Orleans, LA. You can reach him at andrew@downs.ws.

infrastructure. Some businesses base their products on code originating from free software.

Freedom to modify is a pillar of this philosophy. The responsibility accompanying that freedom comes in the form of the GNU Public License (GPL). The GPL states that distributing a product that uses free software within its core or periphery (derivatives of the GPL differ) requires the distributor or vendor to also distribute or make available the product source code so that others may enjoy the same freedom to use and modify.

MAKING MONEY

One important question with regard to both free software and open source software (which by definition is free-of-charge) is "How do you make money?" At first blush it may appear impossible. After all, I or any programmer or hacker with a compiler can download, modify, and build the finished product. Then I can simply give it to my friends and coworkers, right?

Yes, you can if the license permits it. But your next question might be "How do I configure it for my system?" Or "Why did it not install properly on my system?" If you have time and desire you may be able to determine the answers on your own. But businesses typically do not plan for, pay for, or sometimes allow this experimentation to occur on their dime. They prefer to call or email someone else for support.

And there lies the answer to the moneymaking question. The rise in outsourcing encourages firms that specialize in support (including packaging, installation, configuration, and usage issues) of open source and free software products. Businesses can then expense support costs by purchasing a packaged GNU/Linux product from a vendor such as RedHat, Caldera, SuSE, or VA Linux and receive ongoing support from outside the company. This reduces or eliminates the need for in-house expertise, though pockets of knowledge within the company will remain and likely grow.

MICROSOFT

Microsoft's role in the movie is cannon fodder for several anecdotal episodes discussing encounters between free software and open source luminaries and Microsoft managers and the company-at-large. One of the most memorable is Bill Gates' 1976 letter to the Homebrew Computer Club, in which he verbally clubs those developers who used Microsoft's BASIC compiler without paying for it. In contrast, the Windows Refund Day protest in 1999 provided less drama, since Microsoft provided drinks rather than water cannons when the protestors showed up at their building. But the movie is not a Microsoft bashfest. Rather, it illustrates the reasons behind the rise of open source and free software. These people changed the world in response to the proprietary software philosophy, of which Microsoft is the most visible proponent.

APPLE

So where is the Apple and Mac OS connection in all this? The movie discusses free software (the GNU tools) and open

source (Berkeley Unix) products that are familiar territory to Mac OS developers with the rise of OS X. The GNU development tools lie at the center of Project Builder, the development environment that ships with OS X. Berkeley Unix, itself an attempt to bring freedom to the category of operating systems by competing with AT&T's proprietary Unix, has over the years given rise to offshoots that use open source and collaboration as their development philosophy. FreeBSD is one of those offshoots and sits near the core of OS X. (The Mach microkernel resides underneath. Mach is another open source project that receives little air time in the movie but broke new ground in kernel development.)

At the other end of the Mac OS spectrum are technologies such as Aqua and QuickTime that are unlikely to become open source candidates since they are key to differentiating Mac OS from competing operating systems.

In the middle lie extensions to the kernel and BSD subsystems, including Directory Services and Rendezvous. These remain open source projects maintained by Apple and interested developers outside the company.

SUGGESTED READING

I enjoyed this movie, but it may leave some technophiles longing for more. Its treatment of technical issues is light. If you find the movie intriguing or simply want to learn more about the people, ideas, and products involved, here are several books you should consider reading:

- The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, by Eric S. Raymond, 1999, 2001. O'Reilly and Associates, Inc. The essays in this book capture the essence of the open source philosophy and attempt to quantify and explain the "why" behind the "what".
- Free as in Freedom: Richard Stallman's Crusade for Free Software, by Sam Williams, 2002. O'Reilly and Associates, Inc. This book presents Richard Stallman's free software philosophy and history through recaps of interviews with the man himself and other sources.
- Just for Fun: The Story of an Accidental Revolutionary, by Linus Torvalds and David Diamond, 2001. HarperCollins. Half-written by Linus Torvalds, with the balance consisting of interview recaps, this book provides insight into Linus' character and views on life, Linux, and the pursuit of family.
- Open Sources: Voices from the Open Source Revolution, edited by Chris DiBona, Sam Ockman, and Mark Stone, 1999. O'Reilly and Associates, Inc. This contains essays by Stallman, Raymond, Torvalds, Perens, Tiemann and others.
- Rebel Code: The Inside Story of Linux and the Open Source Revolution, by Glyn Moody, 2001. Perseus Publishing. An account of the rise of Linux from a spare-time college project to its dominance today.
- Also, check out the Revolution OS web site at <http://www.revolution-os.com/>.

By Jay Martin Anderson

From Algorithm to Animation: the Sequel

Making QuickTime movies which illustrate abstract processes

THE ORIGINAL

In the November 1993 issue of *MacTech* (volume 9, number 11), I described how I made a QuickTime movie of an algorithm from computational mathematics. In the intervening ten years, QuickTime has grown and matured, the Mac OS has crossed the divide between Classic and OS X, and several generations of IDEs have come and gone.

My interest in making QuickTime movies to illustrate algorithms from computer science and mathematics has not. Consequently, I wish to present an update of the technique used to make animations of algorithms, and apply it to a theorem from beginning calculus which my colleagues challenged me to illustrate this year.

THE PROBLEM

The Department of Mathematics at Franklin and Marshall College includes both mathematics and computer science and frequently enjoys sharing problems in one area with the other. In the spring of 2003, my mathematician colleagues challenged me to construct a way to depict the *Mean Value Theorem (MVT)* which states that, for a suitably "nice" function $f(x)$ on a range $a \leq x \leq b$, somewhere (some $x = c$, $a \leq c \leq b$) the slope of the function, $f'(c)$ is equal to the "rise over run" of the function, $(f(b)-f(a))/(b-a)$.

I chose to take a particular function $f(x) = x^3$ on the range $-1 \leq x \leq 1$, trace along the function showing a line tangent to the curve at each point, and stop when the tangent line had the same slope as the "rise over run," which is 1.

THE PARADIGM

At F&M we have developed a paradigm for constructing algorithm animations. We divide the moviemaking application into three parts: the model, the view, and the controller.

The model contains the algorithm to be visualized. In this example, it is the function and its derivative. The *state* of the model is the value of x , along with the value of $f(x)$ and $f'(x)$.

The view contains the graphical elements necessary to illustrate the model. It is in the view that a single picture of a single state of the model is developed. In our work beginning in 2003, the view is developed with OpenGL. In the view we will draw the graph of the function, the tangent line at every value of x , the line which represents the "rise over run," and any tangent line with the same slope as "rise over run."

The controller does all the initialization and cleanup necessary for the moviemaking application. In particular, it does all the work necessary to pass individual views to QuickTime for the formation of a movie. Most importantly, the controller contains the timing, which signals the model to change its state and the view to generate a new picture. The controller will continue to ask the model to change its state until the value of x has spanned the entire range of interest.

Many movies are divided into "acts" and "scenes," much like a real theater piece. This provides a simple way to modularize the model and the view, and is particularly useful if the animated view of the model is more complex than simply moving an object along a path. For this simple example, acts and scenes are not necessary.

Our movies are also annotated. A view of a state of the model is also accompanied by a few sentences of text which explain what the view shows. The resulting QuickTime movie has one video track, and one text track for each (natural) language necessary. The natural language feature of QuickTime makes it particularly easy to construct movies in English, German, Italian, etc. The text annotation in this movie gives numerical values for x and $f(x)$.

Beginning in 2003, our development platform for algorithm animation is Apple's Developer Tools, including Project Builder and Interface Builder. The application is developed in Objective-C, using Cocoa classes. The model is

Jay Martin Anderson is the Richard S. and Ann B. Barshinger Professor of Computer Science at Franklin and Marshall College in Lancaster, Pennsylvania. Once a physical chemist, he has been teaching computer science since 1978 and making QuickTime movies since 1992. His interests include many aspects of computer graphics, including cartography and computational geometry; opera, theatre, and bicycling. He can be reached at jay.anderson@fandm.edu.

a subclass of **NSObject**; the view is a subclass of **NSOpenGLView**; and the controller is also a subclass of **NSObject**. Since the movie, and not the application which creates it, is the object of this exercise, the graphical user interface is kept as simple as possible, often including only a button "Make Movie" and perhaps some widgets to select among choices for the movie.

The connection between the strictly object-oriented Objective-C and Cocoa classes, and the older QuickTime API is not always elegant. Nonetheless, it is that connection which is perhaps the most significant contribution of this article. Almost every time the controller has to move information between the model or the view and QuickTime, some conversion is necessary.

USING INTERFACE BUILDER

We begin with Project Builder and construct a new project, a Cocoa application. Interface Builder allows us easily to construct a window and view of appropriate size, place whatever GUI widgets we need in the window, construct the model and controller objects, and generate some source code for the model-view-controller system.

Launch Interface Builder by double-clicking the **MainMenu.nib** icon, which is found inside the Resources folder of the Files pane of the project. Make the window an appropriate size, and drag a Custom View onto the window. Make the Custom View be an appropriate size as well. Create a subclass of **NSOpenGLView**, such as **MyOpenGLView**, and make the Custom View be an instance of **MyOpenGLView**.

Drop whatever widgets you need into the window. In this example, we use only a button, labeled **Make Movie**.

Create a subclass of **NSObject** called **MyController** and a subclass of **NSObject** called **MyModel**. Within **MyController**, create an Action method, for example, **createMovie**. Also within **MyController**, create Outlets, which are instances of the classes **MyModel** and **MyOpenGLView**, respectively. Likewise, in the class **MyOpenGLView**, create an Outlet, which is an instance of the class **MyModel**.

Make instances of these classes, and generate code for these classes.

Build connections as follows: *from* any widgets, such as the **Make Movie** button *to* the instance of **MyController**, with the target of the **createMovie** method; *from* the instance of **MyController** *to* the instance of **MyModel**; *from* the instance of **MyController** *to* **MyOpenGLView**; and finally *from* the instance of **MyOpenGLView** *to* the instance of **MyModel**. The connection from the button to the method means that the button sends the **createMovie** message to the controller. The connection from the controller to the model means that the controller can send messages (usually inquiries) to the model, and likewise to the view. The connection from the view to the model means that the view can send messages (again, usually inquiries) to the model.

Long Distance

3.9¢ Per Minute!

Straight 6 second billing increments

Excellent rates on intrastate, intralata/toll calls and international calling with no term contract.

Toll Free (800/888/877/866) service, same low per minute rate for incoming calls.

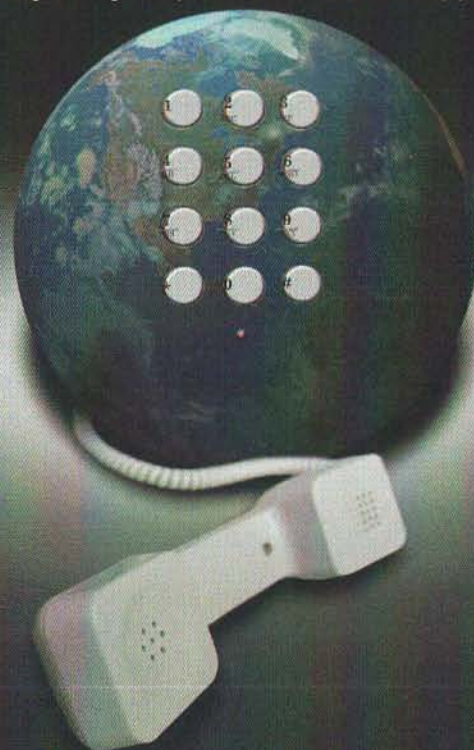
10 cents per minute calling card.

Detailed billing directly from Capsule Communications, a Covista Company.

Quality electronic and telephone customer support.

No monthly billing fee if you sign up for AUTOPAY billing option or if your bill is over \$20.00 each month.

(NOTE: \$1.00 billing fee is charged when your bill is under \$20.00 for all non-Autopay customers.)



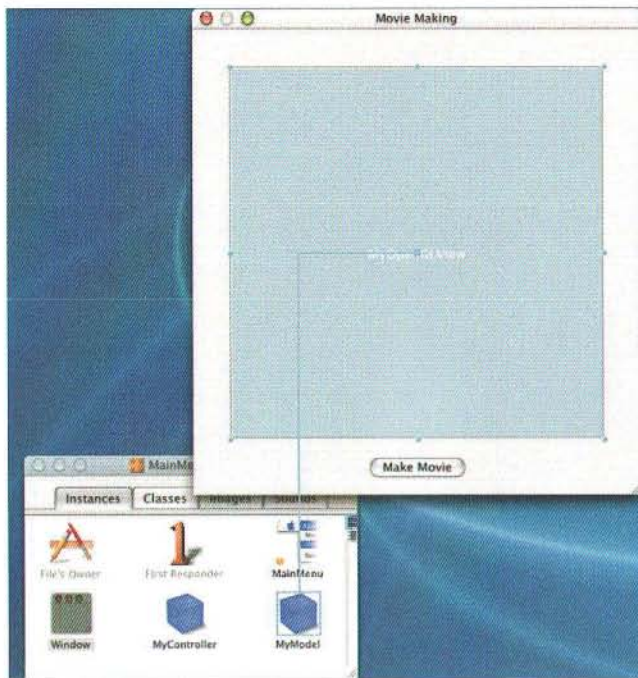


Figure 1. Using Interface Builder. The window contains the view, a subclass of `NSOpenGLView`, and one button. The view is connected to the class `MyModel`, so that the view may inquire the state of the model.

This completes the work with interface builder; save the nib and continue with Project Builder.

THE CODE

The work of managing QuickTime is done with the `createMovie` method of the controller object. This also includes code to provide an interface, albeit awkward, between the Objective-C classes of Cocoa and the "legacy" code of QuickTime. Three points bear special mention.

First, OpenGL draws into a *context*, which is described by a pixel format. QuickTime accepts images from a *pixMap*, which is part of the `GWorld` data structure. The function `CopyNSBitmapImageRepToGWorld` accomplishes the task of copying pixels from the first to the second environment. It is based on the assumption of 32-bit pixels in each structure. Curiously, the copying of pixels requires an `NSBitmapImageRep`, which is four bytes wider than the context.

Second, QuickTime constructs a movie file using an `FSSpec`. The `FSSpec` must be derived from a path or file name in the "classic" Macintosh format, and it must appear as a Pascal string. The function `QTutils_ConvertCToPascalString` serves to convert a C-string to a Pascal string.

Finally, the image drawn by OpenGL is inverted when it is copied to QuickTime. Given that the QuickTime movie is the object of this exercise, it seems simplest to construct the OpenGL



THE LAW OFFICE OF
BRADLEY M. SNIDERMAN

Need help safeguarding your software?

If you're developing software, you need your valuable work protected with trademark and copyright registration, as well as Non Disclosure Agreements.

Then, when you are ready to sell it, you can protect yourself further with a licensing agreement.

I am an attorney practicing in Intellectual Property, Business Formations, Corporate, Commercial and Contract law.

Please give me a call or an e-mail. Reasonable fees.

23679 Calabasas Rd. #558 • Calabasas, CA 91302
PHONE 818-222-0365 FAX 818-591-1038 EMAIL brad@sniderman.com

Hosting Sanctuary for OS X & Xserve

Web Hosting & Development

- MySQL
- Lasso
- Web Objects
- FileMaker Pro

QuickTime Streaming Co-location Services

New Junk Email and Virus Protection from Postini

**Mention this ad and get one month of free hosting
with a 6 month commitment.**

Since 1994 **digital.forest** has provided the most complete Apple Macintosh hosting service in the world. Over the last several years the forest has grown to include service and support for most hosted environments on the internet today. To see how we've grown, please see **www.forest.net** for a summary of our service offerings.

digital.forest



www.forest.net • info@forest.net
877 720-0483 • 425 483-0483

drawings upside-down, and let the pixel copying to QuickTime invert them to the proper orientation.

The code for the class **MyController**, as well as for portions of the classes **MyModel** and **MyOpenGLView**, along with all header files, can be found on the website

<http://eDisk.fandm.edu/~jay.anderson/mactech/example.html>

The model is the simplest unit in this example. The methods `f` and `ff` serve to calculate the function and its derivative. The method `changeState` advances a point along the curve by equal increments in the `x`-direction, and computes the corresponding value of the function and its derivative. The method `changeState` also looks for matches between the slope of the curve and the “rise over run,” and records the matches.

Listing 1: MyModel.m

```

MyModel
A portion of the class MyModel, which encapsulates the algorithm to be illustrated and
its data structures. Shown here are the method changeState, and the methods that
evaluate the function and its derivative. The remainder of the class, and the header file
MyModel.h are found on the website.

@implementation MyModel
- (void)changeState:(int)frNo
{
    if (frNo == 0)
    {
        // beginning frame is blank, with only an annotation
        strcpy(annotation, "MEAN VALUE THEOREM. Compare
rise/run to slope of graph.");
    }
    else
    {
        // as frame number goes from 0 to 200, x goes from -1.0 to 1.0 in 200 steps
        x = -1.0 + frNo/100.0;
        y = [self f:x];
        slope = [self ff:x];
        sprintf(annotation, "x = %5.2f, y = %5.2f, slope =
%5.2f", x, y, slope);
        if (fabs(slope - ror) < 0.01)
        {
            // slope matches rise/run; record this position, append extra annotation
            match = 1;
            xMatch[nMatches++] = x;
            strcat(annotation, "\rSlope matches rise/run: mean
value theorem!");
        }
        else
            match = 0;
    }
}
- (float)f:(float)z
{ return z*z*z; }
- (float)ff:(float)z
{ return 3*z*z; }
@end

```

The view is responsible for drawing the curve, the “rise over run” line, a point and tangent line which moves along the curve as the state of the model changes, and any matches between the slope of the curve and the rise-over-run line. These are simple OpenGL primitives: point, line and line-strip. Colors, the point size and the line width are chosen for viewing ease.

Listing 2: MyOpenGLView.m

The method `drawRect` from the class **MyOpenGLView**; the remainder of the class and the header file **MyOpenGLView.h** are found on the website. This method is invoked when the display message is sent to the view. It draws each frame of the movie.

```

MyOpenGLView
@implementation MyOpenGLView
- (void)drawRect:(NSRect)rect
{
    float x, y, slope;
    short match;

    glClear(GL_COLOR_BUFFER_BIT); // clear the screen
    x = model->x;
    y = model->y;
    slope = model->slope;
    match = model->match;

    /* Draw the components of the frame: the graph of the function, the rise/run line,
the point at (x,y), and the slope of the graph at that point.
*/
    [self drawGraph];
    [self drawRR];
    [self drawPoint:x:y];
    [self drawSlope:x:y:slope];

    /* Draw a tangent line at each place where the slope of the function matches the
rise/run line.
*/
    if (model->nMatches > 0)
    {
        int i;
        for (i = 0; i < model->nMatches; i++)
            [self drawMatch:i];
    }
    // show it!
    glFlush();
}
@end

```

The text annotations are developed within the model, and left in a string. In the controller, this string is copied to the text handler, just as the image is copied to an image compressor. Color, font, and size of text can be chosen when the text handler is constructed.

LIGHTS! CAMERA! ACTION!

The controller runs a simple loop to construct successive frames of the movie. In particular, the controller sends the `changeState` message to the model, and then the `display` message to the view. This image is then copied to QuickTime and compressed. In addition, the controller sends a `getText` message to the model to acquire the annotation to be placed into the text track.

RESULTS

The model-view-controller paradigm has been implemented for making QuickTime movies that illustrate algorithms in Java and in Cocoa/Objective-C. The Cocoa/Objective-C implementation allows the use of OpenGL with QuickTime in a workable, if awkward, marriage of Objective-C classes with “legacy” QuickTime APIs. This opens up abstract moviemaking to the standard of OpenGL, including three-dimensional graphics. The Java implementation, which is not described here, does not require the awkward connection between software components, but does not provide easy access to OpenGL.

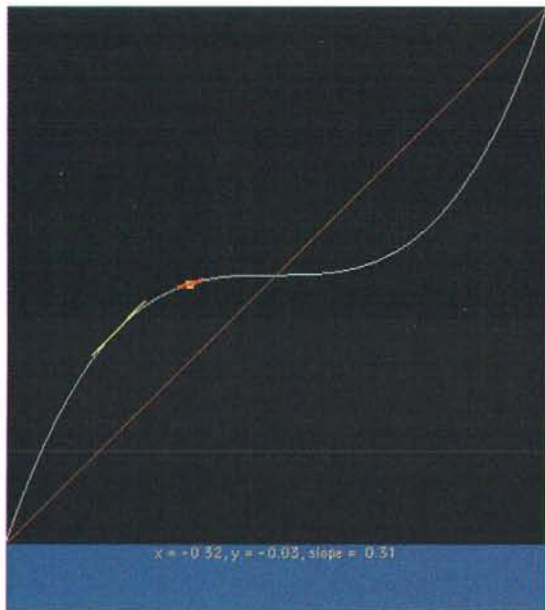


Figure 2. One frame of the Mean Value Theorem movie. The test point has moved from $x = -1.00$ to $x = -0.32$, and has found one spot where the slope of the curve matches the rise/run line. The text track is shown in white on blue beneath the video track.

Besides the sample movie described here, our work in 2003 and beyond includes constructing animations of algorithms from computational geometry, such as the triangulation of a polygon, constructing and searching in kd-trees and range trees, and more. Our recent work includes animations of algorithms for computing the convex hull, for the intersection of many line segments and for recognizing the polygon in which a point lies. Sample movies are posted on our web site (begin at <http://eDisk.fandm.edu/~jay.anderson>) as they become available.

ACKNOWLEDGMENTS

Significant assistance has come from Apple Computer, from whom code to construct QuickTime movies, media, and tracks has been borrowed; and from Big Nerd Ranch and Aaron Hillegass, whose Cocoa programming book, course and advice were enormously useful.

MacTech[®]
M A G A Z I N E

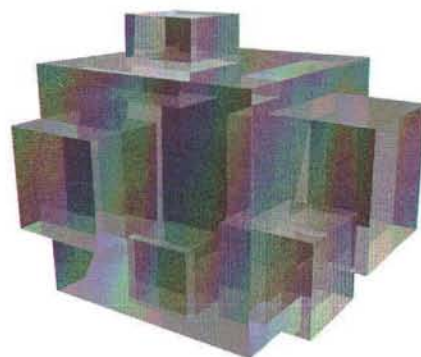
Get MacTech delivered to your door at a price **FAR BELOW** the newsstand price. And, it's **RISK FREE!**

Subscribe Today!
www.mactech.com

to: no-time-to-waste-programmers
subject: version management

select the uncompromising-OS X
only minimalist project mart

VoodooServer 2



www.unisoftwareplus.com

By Brian Shin

Secure File Transfers With Fugu

Review and how to for Fugu

YOUR LAST BITE?

Anybody who speaks Japanese would recognize fugu as the diodon holocanthus, the poisonous blowfish. This delicacy, popular in Japan and the Philippines, contains a poison called tetrodotoxin, 1200 times deadlier than cyanide. Of course this has nothing to do with our Fugu, a wonderful graphical interface to secure file transfer (SFTP) and secure copy (SCP). Fugu's home page is <http://rsug.itd.umich.edu/software/fugu/>. Fugu is freeware and can be downloaded from <http://rsug.itd.umich.edu/software/fugu/download.html>.

PRYING EYES

Our non-toxic Fugu allows us to transfer files with the same ease of use of Fetch but retains the security of command line SFTP and SCP. Regular FTP sends the user name and password in clear text. This means anybody who can sniff your Ethernet packets, can get your user name and password. Using Etherpeek™ and a shareware hex editor, I was able to sniff out my own FTP user and password in just a few minutes;

```
User:
\USER jlpi
card. 5C 01 55
53 45 52 20 6A 6
C 70 69 63 61 72
64 0D. ....
```

```
Password:
\PASS ent
erpris 5D 0E 50
41 53 53 20 65
6E 74 65 72 70 7
2 69 73. e.....
```

As you can see, the hex editor allows me to read the user name "jlpicard" and the password "enterprise". With FTP this easy to hack, you can see why it is important to use SFTP and SCP. Running the same test using Fugu rendered the encrypted user and password completely inaccessible and protected from prying eyes.

KEEP THE DOOR CLOSED

Another benefit from using Fugu is not having to use port 21 (FTP), one of the most common ports used for hacking. Port 21 is subject to constant attack from hackers on the internet. Fugu uses port 22 instead of port 21. This allows you to turn off your FTP servers and close port 21. A search on Google for FTP hacks will leave you stunned as to what hackers are doing with FTP and port 21.

MAKING THE CONNECTION

Servers you connect to need to have SSHD running. In OS X, getting SSHD running is as easy as going into System Preferences, select the Share pane, select the services tab, and turning on Remote Login. For OS X server, you can configure this in the server admin utility. If you are on a different server, go to <http://www.openssh.org/> for the installer.

Making the connection with Fugu is as easy as it gets:



Figure 1. Connecting to a server with Fugu is as simple as entering the IP address and user name.

Brian Shin once worked for us at MacTech, but moved on to run his own company, providing accounting software solutions to Mac run businesses. Between SCUBA dives, he also provides consulting and other programming services. You can reach him at brian@prometheia.com.

NOW RUNNING ON YOUR MAC!



Figure 2. Type in the password and press Authenticate.

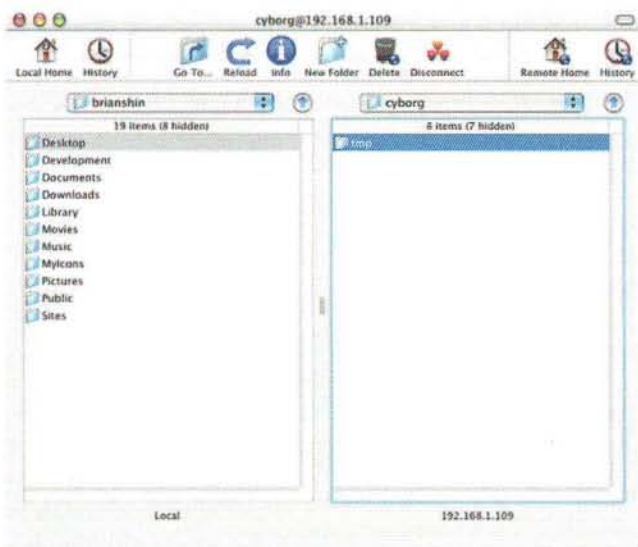


Figure 3. After you authenticate, Fugu will take you to the users home directory.

Once connected to the server, Fugu behaves like just about any other FTP client. The only exception is folder copying. According to the documentation, SFTP does not support folder copying. Fugu circumvents this by turning the folder copy attempt from SFTP into a SCP. Fugu will also ask you to re-authenticate the copy. This works fine but it will not retain the directory information of the copy. For example, you need to copy the directory `../stuff` to your computer at `../Desktop/stuff`. The contents of `../stuff` on the remote computer will all be dumped to `../Desktop`.

Mac OS X



IP*Works! The Only Truly Comprehensive Internet Toolkit

- More Than 30 Components Cover All Major Internet Protocols
- Follows Exact RFC Specifications
- Market-Tested For Over 7 Years
- In Use By Almost All Fortune 500s
- Royalty-Free Pricing

*IP*Works! for Mac OS X (10.0/10.1/10.2) gives you access to a host of new capabilities that will connect your applications to the Internet with unprecedented ease of use, power, and flexibility! You will be able to easily and quickly:*

- program the web: HTTP, WebForm, WebUpload
- call web services: SOAP, XMLp
- transfer files: FTP, TFTP
- send email: SMTP, FileMailer, HTMLMailer
- receive email: POP, IMAP
- read/post news: NNTP
- encode/decode: MIME, NetCode
- access directories: LDAP
- manage networks: SNMP, Whois, Ping, TraceRoute
- build clients and servers: IPPort, IPDaemon
- build packet applications: UDPPort, Multicast
- remote access: Telnet, Rexec, Rshell, RCP

...and a whole lot more! - download your free trial today from www.nsoftware.com!

IP*Works! for
Mac OS X is currently
available as a C/C++
Library (OS X Framework).
Objective-C Classes for
Cocoa and RealBasic
plugins coming soon at
www.nsoftware.com

STAY TUNED!

high quality - competitive rates - 16 years experience - award winning

reliable - high quality - competitive rates - efficient - award winning - qa services - reputable

Full Spectrum Software Development & Testing

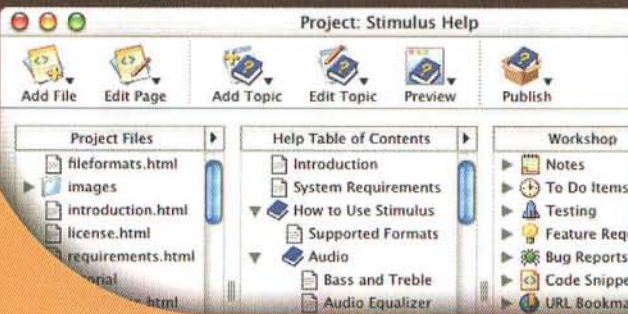
Device Drivers
Porting Plug-ins
TCP/IP
Carbon / OSX
Cross Platform Development

One Bridge Street
Newton, MA 02458
617-965-0029
www.FullSpectrumSoftware.com

competitive rates - 16 years experience - award winning - high quality

HelpLogic

The Help Authoring Solution for Mac Developers



Easily create help systems for your software applications & web sites from a single source.

Save time with the integrated Workshop, TOC Builder, HTML Editor & Page Templates to quickly generate Apple Help, Web-based Help, UniHelp, PDF & more.

SNEAK PREVIEW
www.ebutterfly.com

electric butterfly

CONNECT WITH SFTP

Under the SFTP menu you can get info on files and folders on your system and the remote computer. If you open the console window, you can see command line equivalents to your graphical work and enter direct commands. Notice of all the keyboard commands in the SFTP menu. They make navigation between remote and local directories and panes quick and painless.

CONNECT WITH SCP

SCP transfers work differently than SFTP. Rather than moving files back and forth between computers, SCP transfers 1 file or a directory at a time.

Connections are similar to SFTP but you need the name of the file you want to send or get before-hand.

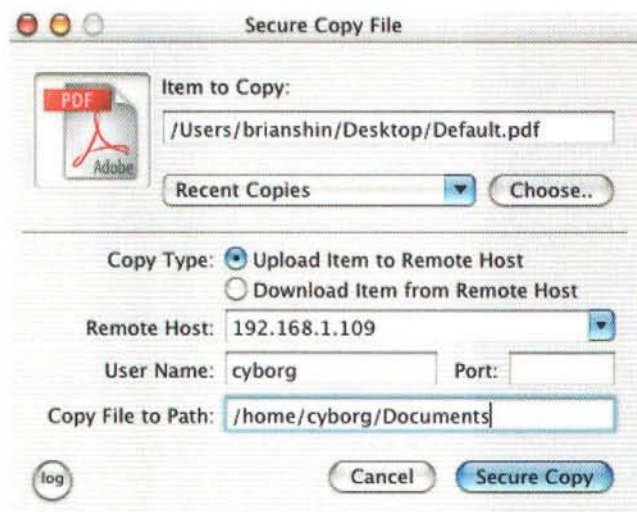


Figure 4. Enter file, upload/download, user id and path.

After you enter this information, a new window will appear to enter the password. If the upload/download transfers correctly Fugu will give you the following message;

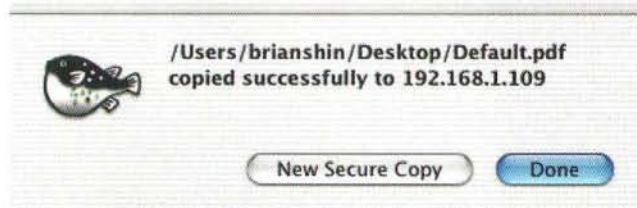


Figure 5. Copy to server successful with option for more transfers.

NO SAFE HARBOR FOR YOUR SHIP?

Now, what happens if you need to connect to a FTP server that does not use SFTP or SCP? You can create a SSH tunnel to protect yourself. From the SSH menu, select new SSH tunnel or press command-T.



Figure 6. New SSH Tunnel window.

Enter the remote host and tunnel host IP numbers. The remote port is usually port 21 and the local port can be any port greater than 1024. Enter your user name for SSH; the port is optional. You will be prompted for your password after you hit the start tunnel button.

Now that your tunnel is up, your FTP connection will be redirected securely through your tunnel host to the FTP server.

MUST HAVE PROGRAM

Fugu is the kind of application that makes OS X so wonderful. It provides a graphical front end to powerful Unix command line tools. The console window gives the user the best of both worlds by allowing users to see what the graphical front is doing and giving the users an opportunity to type in your own commands.

Keyboard lovers will be pleased that every feature you can access with your mouse has a keyboard equivalent. The keyboard shortcuts are listed with every pull down menu. A comprehensive list of them is available from the help menu.

Installation of Fugu is simple, and problem free. The user manual explains features in detail, and provides screen shots of how to use each feature. The manual also has extensive version history and a decent FAQ section.

The source code to Fugu is publicly available and there are German, Dutch, and Japanese localizations. French, Spanish, and Russian are on the way. With publicly available source code, localizations, Fort Knox security, great ease of use, a well-written manual, and top it off with a total cost of free makes Fugu THE essential tool for sharing files over the internet that no user should go without.

It's 3AM on Sunday morning...



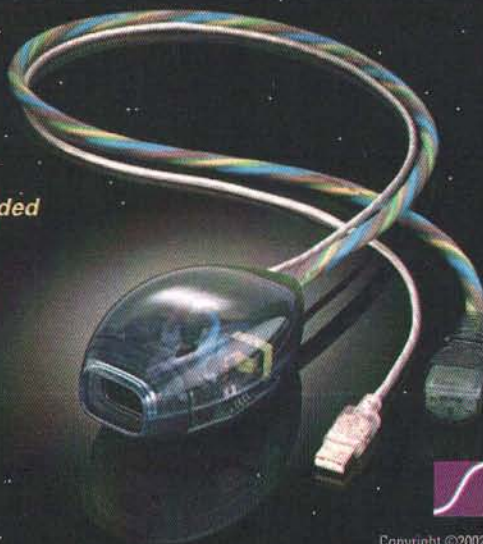
IS THE SERVER STILL RUNNING?

End the
worry with

Kick-off!

The simple reliability
solution for Mac OS 9 and X

- + System-level Crash Detection
patented hardware-software integration
- * Automatic Crash Recovery
system restart or power cycle, as needed
- + Monitor Custom Applications
software developer's kit and plug-ins included
- * Scheduled Restarts
cures memory leaks and fragmentation
- + Restart after Power Failure
even if shut down by your UPS
- * Comprehensive Error Logging
to help you track down problems
- + Simple Installation
just plug in AC cord and USB cable



For all these features *plus* six power outlets controllable by phone tones, schedules and scripts, consider our PowerKey Pro 650 Admin.

www.sophisticated.com

SOPHISTICATED CIRCUITS INC.

Copyright ©2002 Sophisticated Circuits, Inc. Kick-off! and PowerKey are registered trademarks of Sophisticated Circuits, Inc. Mac OS is a registered trademark of Apple Computer, Inc.

By Michael R. Harvey, Reviews Columnist

TiGlide and iGlide

Eliminate the squeak, level the screen

INTRODUCTION

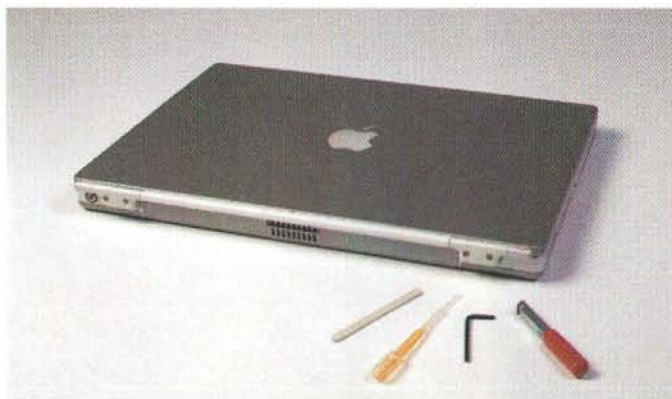
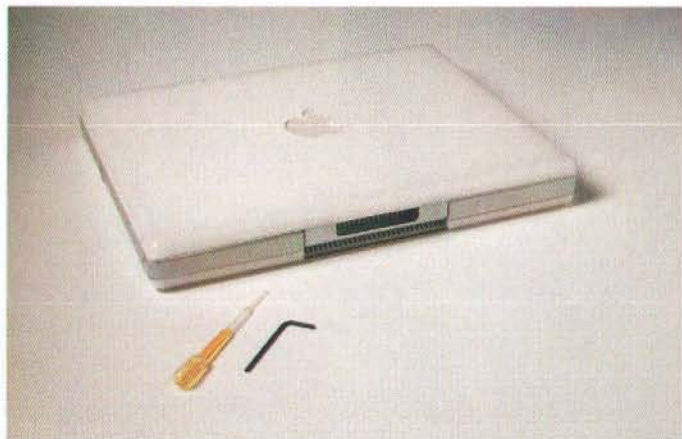
Every now and again, one company will come up with tools that are so simple, yet so necessary, you wonder why seventeen other people didn't think of it before. RadTech is one such outfit. These guys have come up with several great, yet simple, tools designed to protect your favorite mobile computing platform. This time out, we'll be looking at their tools designed to level the screens on PowerBooks and iBooks, as well as smooth hinge operation.

IGLIDE

The kit for smoothing the hinge operation on iBooks comes with a small hex wrench, a vial of metal conditioner, and detailed, yet easy to follow directions. Follow the instructions, and you will shortly find the hinge on your iBook operates smoother than ever. The iBook we tested was very tight, and squeaked something fierce before applying RadTech's changes. Afterwards, operation was smooth and silent. Be careful to follow the directions precisely, though. We were not that diligent, and managed to crack the case on our iBook around one of the hex screws on the underside of the laptop.

TIGLIDE

Like the iGlide, the TiGlide kit comes with all mentioned above. It also comes with a tool to help loosen the hinge bushings in order to smooth operation, as well as shims and a wood pry tool to level the screen. Again, follow the directions



carefully in order to prevent damaging your computer. We performed the adjustments on several Titanium PowerBooks. All saw vast improvement in hinge operation. We were able to level the screen as well, but it would not really hold for reasons we'll get into next.

WHAT'S THE DEAL?

What is it that makes these kits work, and why doesn't the chassis alignment on the TiBooks hold very well? First the main part of the kit, the one thing that makes it all work well, is the metal conditioner. It is a synthetic hydrocarbon that acts as a lubricant, smoothing the hinge operation. It works better than any other oil or grease you could use, and it is safe for the materials that make up the laptop. Next is the problem with keeping the Titanium PowerBook G4's screen level with the chassis. While the process works just fine, there is nothing to help it stay that way included in the kit. RadTech does offer a product they call Wildeepz, display cushions designed to keep the display level as well as keep the screen from touching the keyboard when closed. It would be nice if they were included in the TiGlide kit, instead of as a separate purchase.

CONCLUSION

RadTech has quite a line of useful products designed to extend the life of your PowerBook and iBook computers. The TiGlide and iGlide are two such, and they do their job almost flawlessly. They are well worth the price, \$10 for the iGlide kit, and either \$12.95 or \$15 for the TiGlide (without or with a T-8 Torx driver, respectively). They are more than worth the small cost.

www.radtech.us

***"Without a doubt, the Premiere Resource Editor
for the Mac OS ... A wealth of time-saving tools."***

- MacUser Magazine Eddy Awards

"A distinct improvement over Apple's ResEdit."

- MacTech Magazine

"Every Mac OS developer should own a copy of Resorcerer."

- Leonard Rosenthol, Aladdin Systems

***"Without Resorcerer, our localization efforts would look like a
Tower of Babel. Don't do product without it!"***

- Greg Galanos, CEO and President, Metrowerks

"Resorcerer's data template system is amazing."

- Bill Goodman, author of Smaller Installer and Compact Pro

"Resorcerer Rocks! Buy it, you will NOT regret it."

- Joe Zobkiw, author of A Fragment of Your Imagination

"Resorcerer will pay for itself many times over in saved time and effort."

- MacUser review

"The template that disassembles 'PICT's is awesome!"

- Bill Steinberg, author of Pyro! and PBTtools

"Resorcerer proved indispensable in its own creation!"

- Doug McKenna, author of Resorcerer



Resorcerer® 2

Version 2.0

The Resource Editor for the Mac™ OS Wizard

ORDERING INFO

Requires System 7.0 or greater,
1.5MB RAM, CD-ROM

Standard price: \$256 (decimal)

Website price: \$128 - \$256

(Educational, quantity, or
other discounts available)

Includes: Electronic documentation
60-day Money-Back Guarantee
Domestic standard shipping

Payment: Check, PO's, or Visa/MC
Taxes: Colorado customers only

Extras (call, fax, or email us):
COD, FedEx, UPS Blue/Red,
International Shipping

MATHEMAESTHETICS, INC.
PO Box 298
Boulder, CO 80306-0298 USA
Phone: (303) 440-0707
Fax: (303) 440-0504
resorcerer@mathemaesthetics.com

**New
in
2.0:**

- Very fast, HFS browser for viewing file tree of all volumes
- Extensibility for new Resorcerer Apprentices (CFM plug-ins)
- New AppleScript Dictionary ('aete') Apprentice Editor
- MacOS 8 Appearance Manager-savvy Control Editor
- PowerPlant text traits and menu command support
- Complete AIFF sound file disassembly template
- Big-, little-, and even mixed-endian template parsing
- Auto-backup during file saves; folder attribute editing
- Ships with PowerPC native, fat, and 68K versions

- Fully supported; it's easier, faster, and more productive than ResEdit
- Safer memory-based, not disk-file-based, design and operation
- All file information and common commands in one easy-to-use window
- Compares resource files, and even **edits your data forks** as well
- Visible, accumulating, editable scrap
- Searches and opens/marks/selects resources by text content
- Makes global resource ID or type changes easily and safely
- Builds resource files from simple Rez-like scripts
- Most editors DeRez directly to the clipboard
- All graphic editors support screen-copying or partial screen-copying
- Hot-linking Value Converter for editing 32 bits in a dozen formats
- Its own 32-bit List Mgr can open and edit very large data structures
- Templates can pre- and post-process any arbitrary data structure
- Includes nearly 200 templates for common system resources
- TMPLs for Installer, MacApp, QT, Balloons, AppleEvent, GX, etc.
- Full integrated support for editing color dialogs and menus
- Try out balloons, 'ictb's, lists and popups, even create C source code
- Integrated single-window Hex/Code Editor, with patching, searching
- Editors for cursors, versions, pictures, bundles, and lots more
- Relied on by thousands of Macintosh developers around the world

To order by credit card, or to get the latest news, bug fixes, updates, and apprentices, visit our website...

www.mathemaesthetics.com

By Chris Kilbourn

From Here to Timbuktu

A venerable tool is updated for OS X

ENSCONCED IN ITS EVOLUTIONARY NICHE

In biology, a holotype is the original specimen used to describe a species. In the taxonomy of Macintosh applications, Timbuktu Pro, published by Netopia, Inc., is the holotype for remote desktop screen sharing and file exchange software.

Timbuktu Pro's original copyright dates to 1987, the era of the MacSE and MacII, and network administrators and remote help desks the world over have been singing its praises ever since. Timbuktu Pro provides a remote desktop screen in a window with the ability to either view or control it, a file exchange function, a text chat method and, if you and a remote machine have microphones, an audio intercom feature.

With the ability to remotely control other computers and exchange files with them, Timbuktu Pro can be a huge time saver in solving problems, providing user training and performing software installations and upgrades. Pre-Timbuktu Pro, you actually had to get out of your chair and visit the user or server to perform even the simplest task. In a campus environment where a server or user might be a fifteen-minute walk away, the benefits of being able to control a machine at a distance are apparent.

Unlike other applications with similar longevity, Timbuktu Pro has remained surprisingly lean and consistent over the years. There has been little feature bloat, and if you were a user from years ago, it would be like returning to your home town after an extended period away: things might look a little different, but you would have no problem finding your way around.

I suspect that much of the reason that this is the case is because Timbuktu Pro fills its niche exceptionally well, has been extremely successful and Netopia has wisely decided not to mess with success.

In version 6.0.3, Netopia has squashed a few niggling bugs, added a feature to allow for limiting remote screen color depth in order to goose performance and allows for aliasing the TCP/IP Zones file used by the TCP/IP Scanner function in order to facilitate sharing the file with Netopia's NetOctopus, a software asset manager. With version 6.0.3, Timbuktu Pro on Mac OS X matches the performance and stability of earlier versions.

WHERE TO FROM HERE?

Timbuktu Pro is a great product. I have been using it for over ten years and it has saved me countless hours and an endless amount of frustration in that time. It performs well, is a solid application that plays well with others and does exactly what it claims to do without fuss or bother.

It just about owns its niche, but faces competition from Apple in the form of Apple Remote Desktop for screen sharing and file exchange. As legions of Macintosh programmers will tell you, you know you have a great product when Apple releases its own branded version.

Given all this, it is hard to review a product with such a long, successful track record. Instead of talking about the screen sharing function, (it is great,) the file exchange performance, (functional and reliable,) or dissect the Preferences menu, (still blissfully straightforward and clear,) I will focus on areas where I think that the application could move forward and retain its competitive edge.

THIS ISN'T YOUR FATHER'S MACINTOSH

With Unix under the hood of MacOS X, we have entered a whole new world of file permissions, file ownership, directory navigation and process management. The current version of Timbuktu Pro deals with many of these issues gracefully while adhering to the Mac OS interface. It would be nice, though, to see Timbuktu Pro embrace some Unix power user functions and tune up some interface features to reflect our new operating system environment.

FILE EXCHANGE FUN

The feature of Timbuktu Pro that I have used the most over the years is the File Exchange function. It has saved me a huge amount of time in checking to see what software has been installed on remote machines and by providing a quick check to see if a remote server is up or down without incurring the system overhead of a screen sharing connection.

The trusty user interface of side-by-side directory listings of the File Exchange function has not changed much since the Font/DA Mover pioneered this type of view. It is time for a tune-up and some feature enhancements, though.

Chris Kilbourn is an independent small business, network and web infrastructure consultant. Chris is also the founder of digital.forest, Inc., <http://www.forest.net>, which offers database, application and web hosting services in addition to server colocation. You may contact him at chrisk@forest.net.



Figure 1 - Timbuktu Pro's File Exchange Window

A welcome addition to the user interface would be a directory Favorites menu. I find that when I am transferring files inbound to my computer, they are usually going to one or two places on a regular basis, and having to repetitively navigate to those directories can become a tedious task. OS X's Open and Save dialog boxes have the directory Favorites menu as a core user interface feature, and placing it in the File Exchange window would save keystrokes, mouse clicks and time.

Another minor nit with this interface is the default local directory Timbuktu Pro starts you at the Desktop of the root account. This view should really start at your home directory or allow you to set the default directory to begin browsing in. OS X saves your documents in your home directory's hierarchy by default, so why doesn't Timbuktu Pro? Having to navigate there every time seems kind of silly.

For power users, I would like to see options to toggle viewing file ownership, viewing group membership, the ability to change file permissions, viewing alias/link targets, viewing file size, viewing icons, show/hide hidden files and the ability to filter the list view by file type or name. This may sound like lots of feature requests, but the Unix *ls*, *chmod*, *chown*, *chgrp* and *grep* commands will perform these functions for you. They are just waiting in */bin* and */usr/bin* to be tapped and utilized.

Providing local and remote user and group views along with the ability to adjust the user and group ownership, (*chown* and *chgrp*), of remote files saves a file download/modify/upload cycle when you have a need to adjust remote file ownership.

Using the *ls* command in Unix provides a wealth of file information, and you can expand or narrow the scope of information it provides by utilizing option flags. Providing a clickable unix-style file permissions listing, (i.e. - *rw-r--r--*), where you could click on the permissions you wanted to enable or disable would save having to go through the process of downloading the file, changing the file permissions and then uploading it to the remote computer.

Having a directory filename and file type filter, (*grep*), would speed the location of an errant .html file that was misplaced in a graphics directory where there are hundreds of files to sift through. Timbuktu Pro's search feature is very fast, but does not allow you to search within folders.

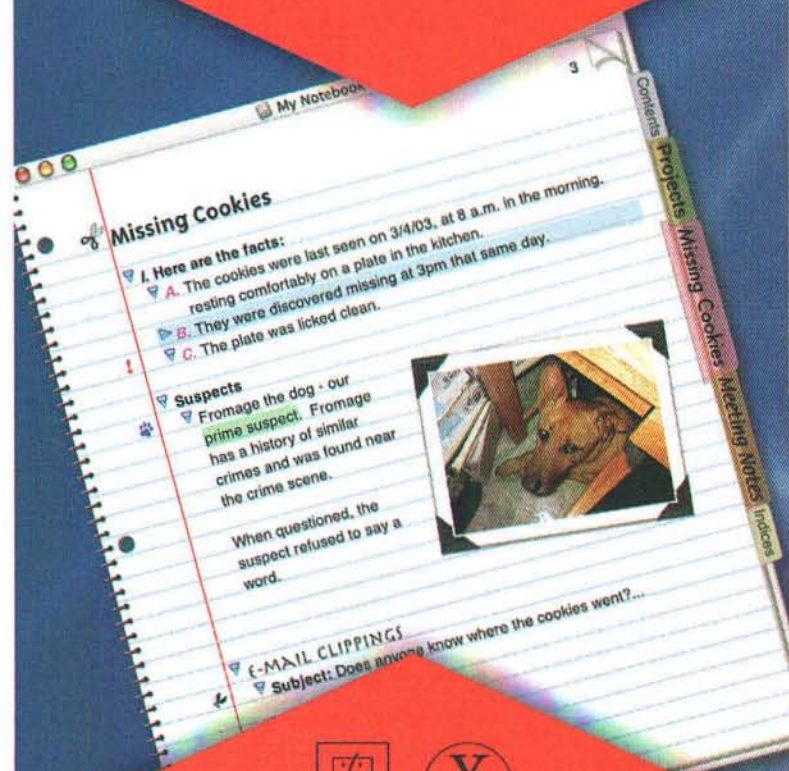
By being able to toggle these Unix commands in Timbuktu Pro's File Exchange window, the exchange feature would be transformed into a sophisticated remote file management tool.



NOTEBOOK™

Getting organized just got easier

- Intuitive outlining
- Customizable notebook interface
- Annotation with stickers, keywords, & highlighting
- Dynamic indexing
- Super-Find searching
- Drag-and-drop Page Bundles
- Inter-application Clipping Services
- Direct import from digital cameras
- Rich image handling
- Media Frames
- Export to XML
- Online help



circusponies.com

© 2003 Circus Ponies Software, Inc. All Rights Reserved. Circus Ponies, the Circus Ponies logo, NoteBook, and the NoteBook logos are trademarks of Circus Ponies Software. All other trademarks mentioned belong to their respective owners.

Providing the option of selecting these features as general preferences coupled with the ability to create filter or view preference sets would provide power users more control over file manipulation. Filtering out information from the list view might also provide a significant speed boost to the display of files in the File Exchange window. For example, if I do not want to see icons or file size, those are two less pieces of data that need to be fetched and displayed.

STUPID SHELL TRICKS

Old-school Unix system administrators were constantly searching for the most efficient combination of shell commands with the least amount of keystrokes in order to accomplish a given task. The command-line interface of the shell was continuously expanded over the years to allow all sorts of various and sundry file manipulation tools that provided sophisticated features, assuming of course, you knew they were there and knew how to apply them to the task at hand.

One of the reasons why you might transfer a file with Timbuktu Pro is to do something with it. You might need to edit it, or email it off to someone, or merge it into another file you are working on.

I know that I cannot count the number of times I have had to download a file from a web server that someone else created, then edit it, and then return it back to the server. During this process, I have had to switch from Timbuktu Pro to the application I am editing the file in and then locate the file before I could begin work.

A feature I would like to see in order to help alleviate some of this hassle is the ability to send a file directly to an application. (Crusty sysadmins will recognize this as the shell pipe feature.) Copying the file to your machine and then having it open up in the targeted application would save a fair amount of effort and needless clicking around.

Borrowing a feature from Apple's Safari, marking a directory location with snap back would complete the process when it was time to return the file. Timbuktu Pro does allow you to save connection documents which will place you in the directory you were browsing, but providing menu access via bookmark-type structure would go a long way towards making server farm administrators and help-desk technicians very happy.

OH, THAT FILE GOES THERE

One File Exchange feature I have always coveted in the exchange function is the ability to move a file on a remote computer. The only way to do this currently is to copy the file to your local computer, then re-copy the file to the proper directory on the remote computer, and then delete the remaining extra files at each end.

This has always been a bother, especially when moving directories around. Some might argue that to move files or directories around, you should use the Control function. My counterpoint to that argument is that can take much longer to move files around on a remote computer via the control function if that computer is heavily loaded with processes or at the end of a low or constrained bandwidth connection.

In the server environments I have worked in with Timbuktu Pro, I have dropped files into directories by accident when the

screen redraw rate slowed down during a Control session. With the File Exchange function, you always know where you are at, directory-wise.

DO YOU SEE WHAT I SEE?

The Control feature in Timbuktu Pro is the meat and potatoes of the program. It allows you to remotely dismiss a pesky dialog box, perform training or troubleshooting for remote users and view how many hits your web server is receiving when it is sitting in a data center on the other side of the planet.

As before, this feature has been rock-solid over the years and it is difficult to say much about it other than the fact that this feature works, it works well, and the screen redraw performance is good. In constrained bandwidth situations, you will encounter jerky remote mouse pointer behavior and slow screen updates, but this is no fault of Timbuktu Pro. A screen redraw meter might be a nice addition to have in situations like that to confirm that it is the redraw that is slow, not your local machine.

For all its success in screen sharing, I do have one major nit to pick about the Control function environment, and another feature I would like to see enabled.

BACK TO THE FUTURE

Try this thought experiment: imagine you are a brand-new Macintosh user that has only ever used Mac OS X. Now, initiate a screen-sharing connection in Timbuktu Pro. Now guess how long it will take you to figure out the window control features.

That's right, there are no Aqua controls in the Look or Control windows of Timbuktu Pro.



Figure 2 - Timbuktu Pro's Control Window Controls

This may seem like a small issue to bring up, but software is hard enough to use as it is, and mixing control interfaces on a user violates user interface guidelines big-time. For a professional's view on violating user interface guidelines, check out <http://www.asktog.com>. To paraphrase Tog, it is a bad, bad thing to do.

There must be legacy code issues in Timbuktu Pro that has prevented Netopia from adopting Aqua interface guidelines in the Look and Control windows; at least, I hope that's the reason. Here's hoping they sort those issues out and are able to bring consistency to the windowing environment in the next release.

COMMAND AND CONTROL

Raise your hand, if while using Timbuktu Pro in Control mode, you have inadvertently closed a remote window or quit a remote application by mistake. Raise your other hand if in doing this, it has caused short-term chaos for you to sort out.

I'm the guy in the corner waving both hands above his head like he is trying to flag down a 747.

Number one on my feature request list for the entire application is the ability to suppress the transmission of command-key combinations to a remotely controlled computer via some sort of toggle switch, with the default being suppression and with the ability to override it on a per-session basis.

Having been the victim of my own fingers, usually when low on sleep, in quitting remote server applications, it would be nice to have this safety feature to prevent user error.

A HERETICAL THOUGHT

Software publishers and software users have reached an accommodation when it comes to features: feature requests are turned into coded features if it is cost-effective to implement them. The marketing and business calculus that drives this dynamic tends to be fairly clear-cut, and has provided us with software that does all sorts of things we need it to do, and some features that we perhaps do not use, but that a significant fraction of other paying customers do.

When this dynamic breaks down and product managers approve every feature requested, we end up with applications that become huge, lumbering beasts that try to be everything to everyone, (Word 6 anyone?). Timbuktu Pro, thankfully, has always been a lithe application free of features that leave you scratching your head asking, "Who the heck needs to do that?"

Operating systems change and the applications bundled with them change over time. Years ago, Apple shipped microphones with computers and monitors to encourage multimedia uses of the machines but they no longer do this. Text chat programs for the Macintosh used to be difficult to install and interoperated poorly, if at all. With the advent of iChat, this is no longer the case.

Lack of built-in microphones and the inclusion of iChat with OS X make the Intercom and Chat functions of Timbuktu Pro mostly superfluous and they should be purged from the application. Why burden the code base with functions that are either difficult to utilize, (by having to buy a microphone,) or are better implemented in another application (iChat)?

THE BOTTOM LINE

Timbuktu Pro performs as advertised, without quirks or crashes for screen sharing and file exchange. Low or constrained bandwidth situations will cause noticeably jerky screen redraw and slower file transfers, but that cannot be blamed on the application. The interface is straightforward, with its features and options easily accessible via menus and key commands.

Version 6.0.3 brings Timbuktu Pro for OS X into the solid performance and stability realm that the application enjoyed under the Classic MacOS. This reviewer hopes that future releases of the application will take advantage of the Unix underpinnings of OS X and make shell commands for file manipulation available when utilizing the Exchange function.

Timbuktu Pro 6.0.3 for MacOS X is a deservedly esteemed application that should be an integral part of your software library if you have a need for remote screen sharing and file transfer.

New PrimeBase 4.2 Replication Server

Check out the fully programmable Replication Server

- Bidirectional Updates supported
- Update 3rd-party DBMS
- Send Emails
- Post/get Data to/from Websites

**SQL-Runtime Plugin
for REALbasic**
\$ 499,-

All PrimeBase Server Software

- SQL-Runtime Libraries available
- SQL Database Server
- Application Server
- Replication Server
- Open Server

Available on the most popular platforms

- Completely cross-platform
- Full-text searching and indexing
- Mac OS & OS X
- Linux
- Solaris
- IBM AIX
- all Windows platforms

 **PRIMEBASE**

SNAP Innovation GmbH
Altonaer Poststraße 9a
D-22767 Hamburg / Germany
www.primebase.com

e-mail: info@primebase.com

Fon: ++49 (40) 389 044-0

Fax: ++49 (40) 389 044-44

by Tim Monroe

Studio 54

Developing QuickTime Applications with AppleScript Studio

INTRODUCTION

In the previous QuickTime Toolkit article, we started building a QuickTime-savvy application using AppleScript Studio. In this article we'll finish up.

Setting Up the Menus

Let's turn to ScripTeez' menus. The Application menu is the easiest to configure, since we simply need to change the name of the application to "ScripTeez" in four instances, as shown in **Figure 13**. I've also set the keyboard shortcut for the "Hide Others" item to be Command-Option-H, as dictated by the Aqua Human Interface Guidelines.

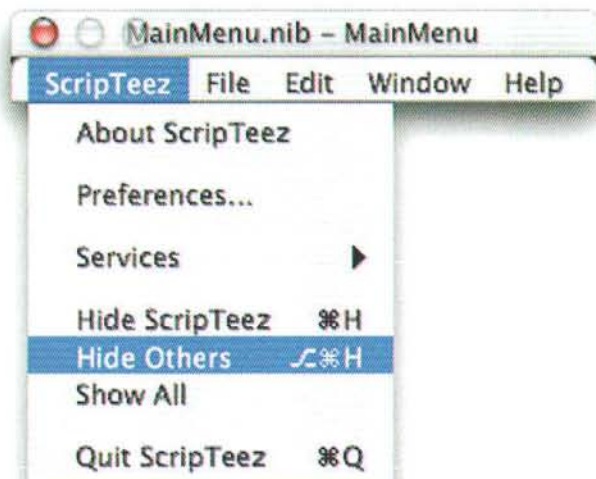


Figure 13: The Application menu nib

In the Edit menu, we need to remove the items that don't apply to movies (the Find and Spelling menu items) and add the "Select None" item, as shown in **Figure 14**.

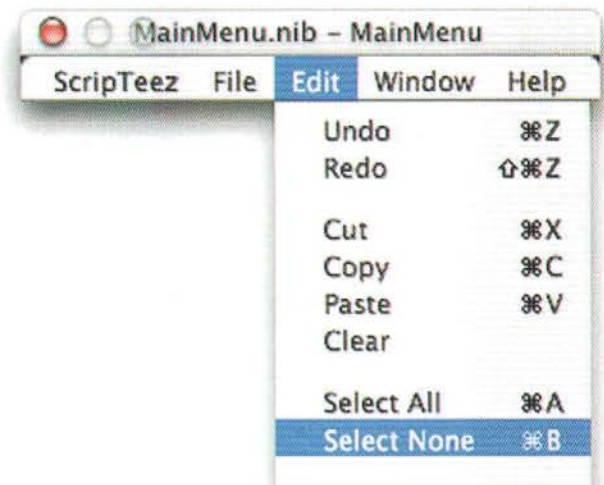


Figure 14: The Edit menu nib

All these items are handled automatically by Cocoa (and in particular, by the **NSMovieView** instance in our movie window), except for the one we just added. In this case, we need to attach an AppleScript event handler. As before, select the "AppleScript" panel in the Info window and then check the "choose menu item" and "update menu item" handlers, as in **Figure 15**. Skeletal handlers are automatically added to the specified script file (that is, **ScripTeez.applescript**); we'll add code to those handlers later.

Tim Monroe is a member of the QuickTime engineering team. You can contact him at monroe@apple.com. The views expressed here are not necessarily shared by his employer.

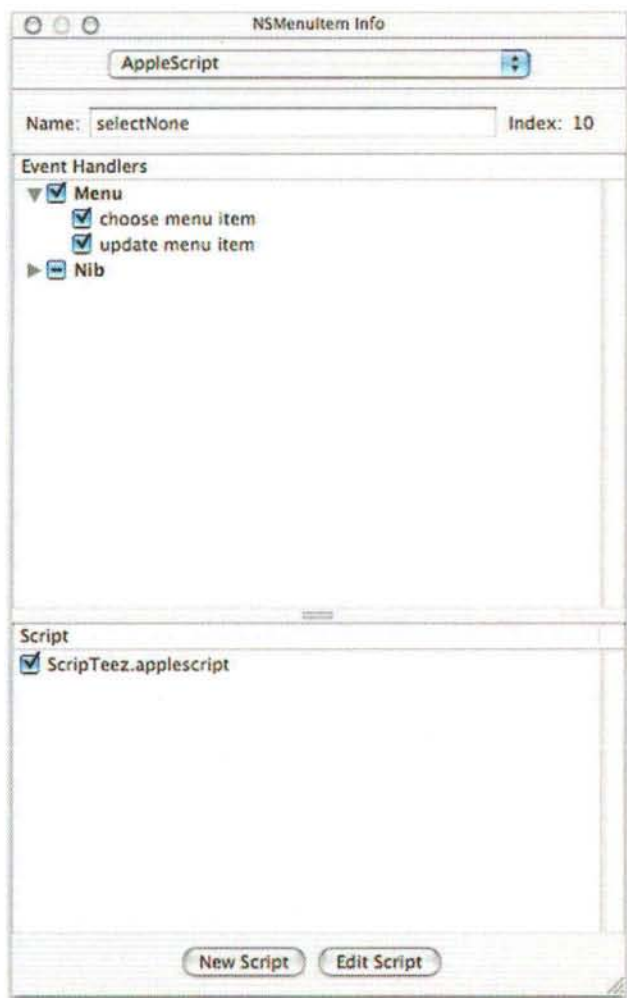


Figure 15: The Select None menu event handlers

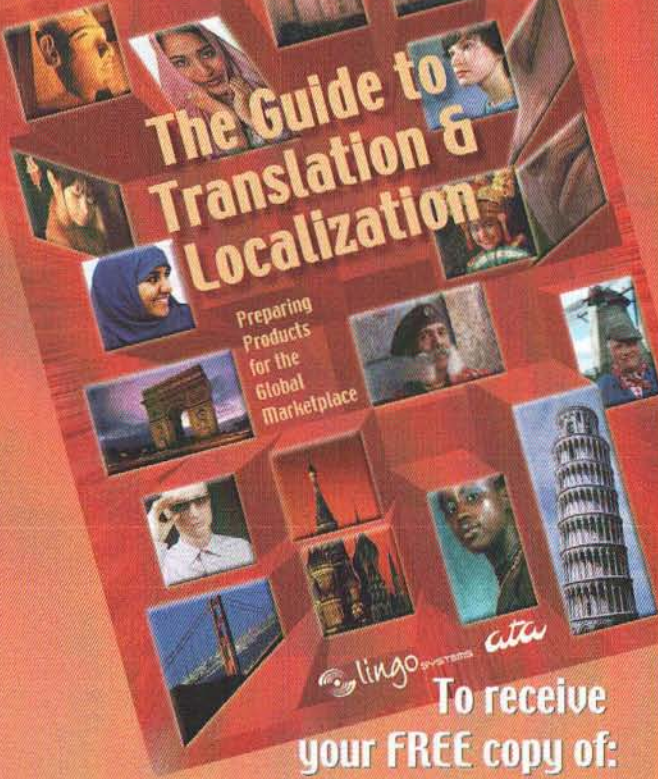
Let's add one more menu to ScripTeez, a Movie menu that allows us to select a looping mode for the movie in the movie window. **Figure 16** shows the updated main menu nib.



Figure 16: The Movie menu nib

As you'd guess, we need to attach AppleScript handlers to adjust and handle these menu items. **Figure 17** shows the Info

Classic or Cocoa Applications? We Localize Them All!



To receive
your FREE copy of:
**The Guide to Translation and
Localization – Preparing Products
for the Global Marketplace**

Call: 1-800-878-8523 Email: info@lingosys.com
Fax: 1-503-419-4873 Or visit: www.lingosys.com

**We focus on what matters most: meeting your
needs and providing excellent customer service.**

- Translation
- Desktop Publishing
- Project Management
- Localization
- Engineering
- Quality Assurance



15115 SW Sequoia Pkwy. #200 Portland, OR 97224

Macworld

 **reativePro**
Conference & Expo.

WM



July 14–18

New York The Javits Center

Conferences July 14–18, 2003

Expo July 16–18, 2003

Innovate. Motivate. Create!



Flagship Sponsors

Macworld

Macworld.com



MacCentral

Platinum Sponsor

HIGH-END
COMMUNICATIONS

Gold Sponsor

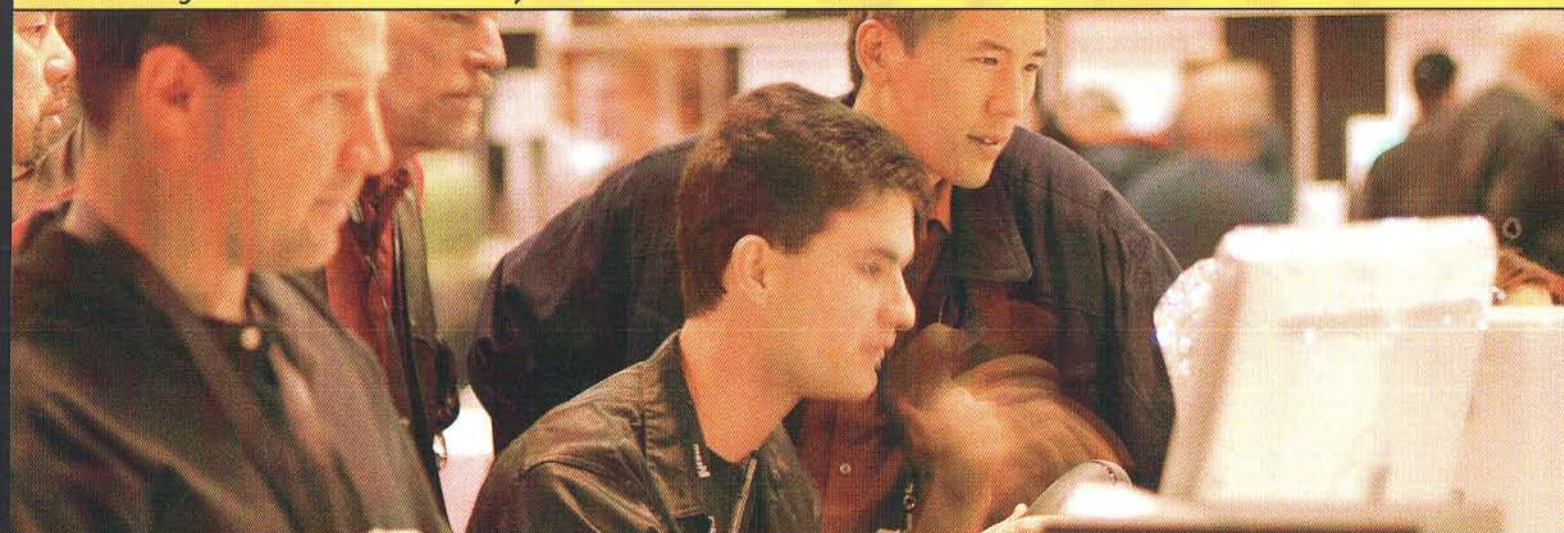
Printing
ForLess.com
YOUR ONLINE RESOURCE FOR QUALITY PRINTING

© 2003 IDG World Expo. All rights reserved. All other trademarks contained herein are the property of the respective owners.

Macworld CreativePro Conference & Expo is a unique series of conferences, focused on technology for the creative arts, including design and publishing, video and music.

Designed for creative professionals and prosumers — everyone who wants to immerse themselves in the creative arts — it's the answer to the creative community's demand for new insights.

Register online with Priority Code: A-MTJ



Your creative arts connection at

www.macworldexpo.com

**1 & 2-Day Tutorials focus
on topics such as:**

- DVD Studio Pro
- Mastering Digital Photography
- Intermediate Dreamweaver
- Desktop Publishing
- QuarkXPress
- Photoshop
- Mac OS X

**Level 1 & 2 Conference sessions
will cover topics such as:**

- Professional & Prosumer Video
- Expert Design
- Advanced PDF
- Preflighting
- Flash
- Color Management
- Editing Techniques for Final Cut Pro, Final Cut Express and iMovie
- How to Set Up a Home Recording Studio
- Music Editing on Your Mac

window for the third item in the Movie menu, the “Palindrome Looping” item. Notice that the name of the item is “palindromeLooping”.

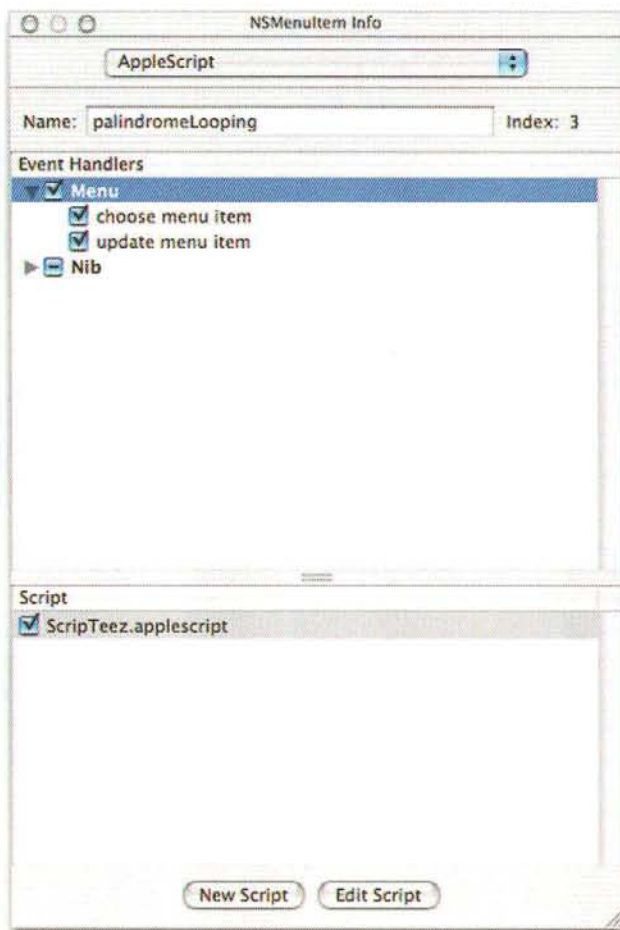


Figure 17: The Palindrome Looping menu event handlers

Adjusting the Project Settings

Before we launch into writing code to handle these events, we need to make a couple of final adjustments to our project. We need to add the QuickTime framework to the project, and we need to specify the kinds of files that our application can open.

To add the QuickTime framework, simply select “Add Frameworks...” in Project Builder’s Project menu and then choose the file “QuickTime.framework”. It will be added to the list of linked frameworks.

To specify the kinds of files our application can open and hence what kinds of files should be selectable in the file-opening dialog box (displayed at application launch time), select the “Edit Active Target” item in the Project menu. Click the “Document Types” item on the left-hand side and add the desired document types. **Figure 18** shows our document types

settings. We want ScripTeez to be able to open QuickTime movie files and Flash files.

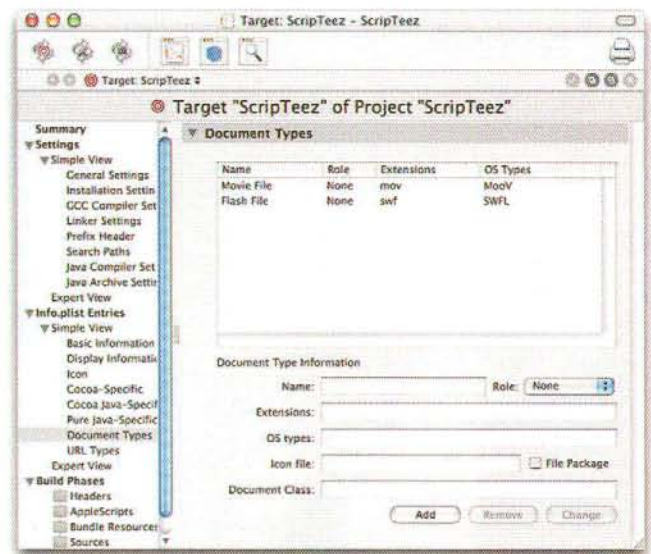


Figure 18: The openable file types

APPLESCRIPT STUDIO MOVIE CLASSES

Now it’s time to write some code to load a movie from a movie file and to handle the menu items we’ve added to the default menu bar. Recall that the only thing we added to the default empty application window was a view of type `NSMovieView`, which we named “movieView”. This name allows us to target AppleScript actions at that movie view. For instance, in the awake-from-nib handler, we might set a local variable `theMovieView` to point to that view like this:

```
set theMovieView to the movie view "movieView" of theObject
```

(Recall that the awake-from-nib handler is passed the object that’s being awakened; in this case, it’s the movie window.)

But what vocabulary can we use to manipulate the movie view? To find this out, we can double-click the item labeled “AppleScriptKit.asdictionary” in the project window (see **Figure 4** again). Expand the item labeled “Control View Suite” in the left-hand column, and then expand the Classes item. We’ll see a couple dozen view types, including “movie view”. If we click on “movie view”, we’ll see the list of properties shown in **Figure 19**.

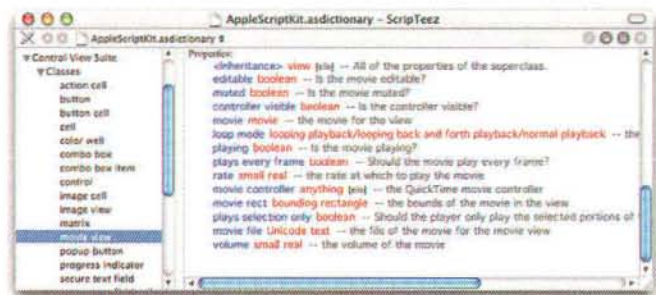


Figure 19: The movie view properties

This list shows us the built-in properties of movie views currently supported by AppleScript Studio. For instance, we can get and set the movie volume, the looping state, and the playback rate. We can get (but not set) the movie controller identifier. We can also get and set the movie associated with the movie view. So, we might set the movie to palindrome looping like this:

```
set the loop mode of theMovieView to ↵
    looping back and forth playback
```

(The character “↵” is AppleScript’s line continuation character; we can insert it into a script by typing Option-L; this allows very long statements to occupy several lines in our script files.)

Loading a Movie from a File

ScriptTeez, you’ll recall, supports only one movie window. We’ll display the standard file-opening dialog box at application launch time, to elicit a movie file from the user. We can display that dialog box and get the full pathname of the selected file with this simple command:

```
set theMoviePath to choose file
```

Then we can assign the movie in that file to the movie view like this:

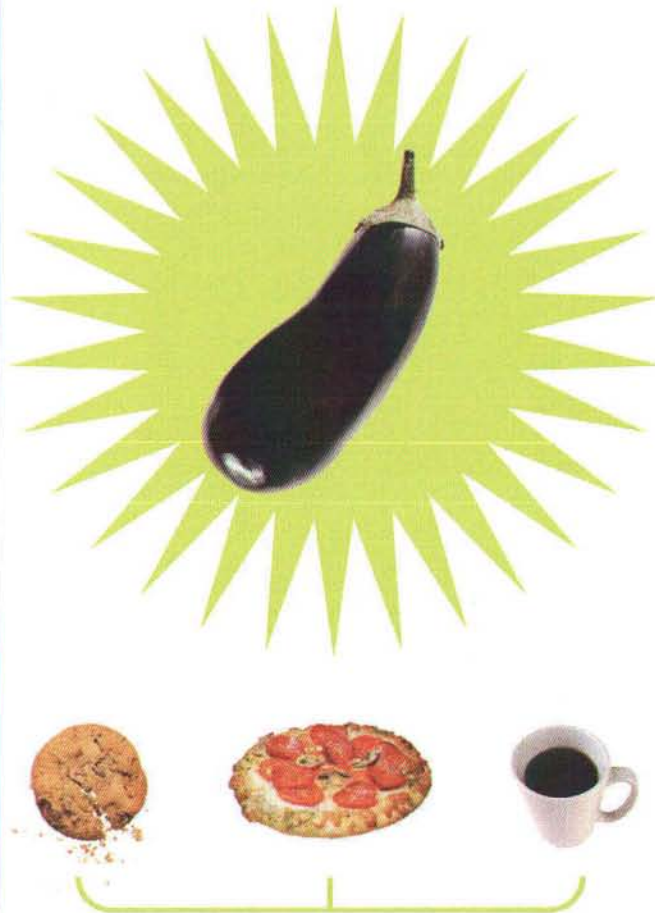
```
set the movie of theMovieView to load movie theMoviePath
```

Setting the Size of a Movie Window

If you look back at **Figure 7**, you’ll see that the “Visible at launch time” check box in the list of movie window attributes is unselected; this is because we don’t want the movie window to be visible while the file-opening dialog is displayed. It’s also because, before we display the movie window to the user, we want to adjust the size of the movie window to exactly contain the movie at its natural size and the 20-pixel border on all sides of the movie view.

The only problem is that AppleScript Studio does not (as far as I can determine) include any built-in method for getting

The newest addition
to every **GREAT**
developer’s diet.



Easy to learn and use
Complete more testing faster
Test any application on any system
Allows testing of remote systems
Captures test results visually

eggplantTM
Test any application on any system.

www.macegg.com

Redstone
SOFTWARE

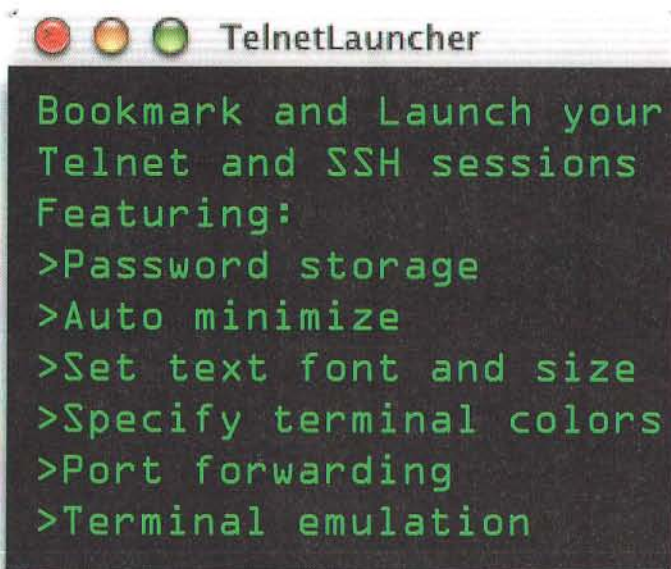
the natural size of a movie. The `movie rect` property returns the current size of the movie rectangle, which will just be the size of the movie view as contained in the nib file once we've assigned the movie to the movie view. Fortunately, AppleScript Studio supports an easy way to call code written in other languages, using the `call method` command. In ScripTeez, we'll need to use this command twice, first to get the natural size of a movie and second to handle the "Select None" menu item.

Let's look at the menu-handling task first, since it's somewhat simpler than the movie-sizing task. When the user chooses the "Select None" menu item, we'll execute this line of script:

```
call method "selectNone:" with parameter theMovieView
```

This `call method` command tells AppleScript Studio to look for an Objective-C method named "`selectNone:`" and to call it, passing as its single argument the value of the variable `theMovieView`.

When we issue the `call method` command, we can specify the class whose method is to be called. For simplicity, however, we'll implement the `selectNone:` method (and the `movieWindowContentRect:` method, which we'll encounter in a moment) as categories on the `NSApplication` class.



This and much more available from...

piDog Software

SimpleKeys - piPop - DockSwap - ScreenShot Plus

www.pidog.com

Info@pidog.com

To begin, let's add two new files to the ScripTeez project; let's call them `ScrTzMethods.m` and `ScrTzMethods.h`. **Listing 10** shows the file `ScrTzMethods.h`.

Listing 10: Declaring a category on NSApplication

```
ScrTzMethods.h
@interface NSApplication (ScrTzMethods)

- (NSRect)movieWindowContentRect:(NSMovieView *)movieView;
- (void)selectNone:(NSMovieView *)movieView;

@end
```

The file `ScrTzMethods.m` contains the actual implementation of the `ScrTzMethods` category. **Listing 11** shows our definition of the `selectNone:` method.

Listing 11: Selecting none of a movie

```
selectNone
- (void)selectNone:(NSMovieView *)movieView
{
    MovieController mc = NULL;
    TimeRecord tr;

    mc = (MovieController)[movieView movieController];
    if (mc != NULL) {
        tr.value.hi = 0;
        tr.value.lo = 0;
        tr.base = 0;
        tr.scale = GetMovieTimeScale(
            [[movieView movie] QTMovie]);
        MCDoAction(mc, mcActionSetSelectionDuration, &tr);
    }
}
```

This is easy stuff that we've seen before. We retrieve the movie controller identifier from the movie view object, fill out a time record appropriately, and then call `MCDoAction` with the `mcActionSetSelectionDuration` action. Notice that we do not return a value to our caller.

Listing 12 shows our implementation of the `movieWindowContentRect:` method. As with `selectNone:`, it takes the movie view as the single input parameter. We retrieve the movie and movie controller identifiers, call `GetMovieNaturalBoundsRect` to get the natural size of the movie, and then adjust the rectangle to contain the movie controller bar (if it's visible) and the 20-pixel border on all sides of the movie view. The rectangle we pass back to the caller contains the desired size of the entire content region of the movie window.

Listing 12: Getting a movie window's size

```
movieWindowContentRect
- (NSRect)movieWindowContentRect:(NSMovieView *)movieView
{
    Rect rect = {0, 0, 0, 0};
    Movie movie = NULL;
    MovieController mc = NULL;

    movie = (Movie)[[movieView movie] QTMovie];
    mc = (MovieController)[movieView movieController];
    if (movie != NULL)
        GetMovieNaturalBoundsRect(movie, &rect);
}
```

```

if (MCGetVisible(mc) == 1)
    rect.bottom += kControllerBarHeight;

return NSMakeRect(0, 0,
    (rect.right - rect.left) + (2 * kMovieWindowBorder),
    (rect.bottom - rect.top) + (2 * kMovieWindowBorder));
}

```

What does our AppleScript call to `movieWindowContentRect:` look like? As with `selectNone:`, we want to pass the movie view `theMovieView` as a parameter. The key difference is that we need to capture the result of the method call, which we can do by copying that result to a local list of values, like this:

```

copy (call method "movieWindowContentRect:" ^
    with parameter theMovieView) to ^
    (theIgnoreLeft, theIgnoreTop, theMovieWindWid, ^
        theMovieWindHgt)

```

We are interested only in the third and fourth items in the `NSRect` structure, which are the desired width and height of the movie window content region. Once we've got those values, we can determine the size and location of the movie window fairly easily. **Listing 13** shows our complete calculation here.

Listing 13: Setting a movie window's size

```

set theTitleBarHgt to 20
on awake from nib

```

```

copy (call method "movieWindowContentRect:" ^
    with parameter theMovieView) to ^
    (theIgnoreLeft, theIgnoreTop, theMovieWindWid, ^
        theMovieWindHgt)
copy the bounds of the theWindow to ^
    (theWindLeft, theWindBottom, theWindRight, theWindTop)
set the bounds of the theWindow to ^
    (theWindLeft, theWindTop - (theMovieWindHgt + ^
        theTitleBarHgt), theWindLeft + theMovieWindWid, ^
        theWindTop)

```

Setting the Title of a Movie Window

One task remains to be performed in the awake-from-nib event handler; we need to set the title of the movie window to the basename of the movie's pathname. (The basename is the portion of the pathname that follows the rightmost path separator.) These three lines of AppleScript will do the job:

```

set the text item delimiters to ":"
set theFileName to (theMoviePath as string)
set the title of the theWindow to ^
    the last word of theFileName

```

We can now make the window visible:

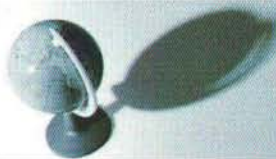
```

show the window of theMovieView

```



effigent, Inc.



We deliver efficient and intelligent business solutions

Providing Server and Client enterprise applications for the Mac platform

Services

- Application Development
- Conversions to Mac
- Testing Services
- Maintenance & Support

Solutions

- E-Commerce
- Education Portals
- Knowledge Management
- ERP Implementations

Technology

- Cocoa
- Mac OS X
- WebObjects & J2EE
- Quicktime & Streaming



info@effigent.com

www.effigent.com

Contact Us: 760-723-4800

Copyright 2003, effigent, Inc. All rights reserved.

MOVIE PLAYBACK

At this point, the user has selected a movie file using the file-opening dialog box; we've loaded the movie in that file into the movie view in the movie window and adjusted the initial size of the movie window as appropriate to display the movie at its natural size. AppleScript Studio, in concert with the relevant Cocoa classes, handles all subsequent user actions like moving or minimizing the window, starting and stopping the movie, editing the movie, and so forth. With very little AppleScript code indeed, and with just a small detour into Objective-C, we've got a fully-functioning movie playback application.

We need to intercede here only to handle the menu items that we added to ScripTeez, namely the "Select None" item and the three looping state items in the Movie menu.

Manipulating a Movie's Looping State

As we saw earlier, a menu item can have two handlers associated with it, one that's called when the state of the menu needs to be adjusted (or "updated") and one that's called when the menu item is actually selected. The update handler is called before the menu item is displayed to the user; typically this occurs when the user clicks somewhere in the menu bar. **Listing 14** shows the complete update handler for our custom menu items.

Listing 14: Adjusting the menus

```
on update menu item theObject
    set theMovieView to movie view "movieView" of ¬
        front window
    set theLoopMode to loop mode of theMovieView

    set enableItem to 1
    set checkItem to 0

    if name of theObject is "selectNone" then
```

```
        if editable of theMovieView is equal to true then ¬
            set enableItem to 1
        else if name of theObject is "noLooping" then
            if theLoopMode is equal to normal playback then ¬
                set checkItem to 1
            set state of theObject to checkItem
        else if name of theObject is "normalLooping" then ¬
            if theLoopMode is equal to looping playback then ¬
                set checkItem to 1
            set state of theObject to checkItem
        else if name of theObject is "palindromeLooping" then ¬
            if theLoopMode is equal to ¬
                looping back and forth playback then ¬
                    set checkItem to 1
            set state of theObject to checkItem
        else
            set enableItem to 0
        end if

    return enableItem
end update menu item
```

Notice that we enable the "Select None" menu item only if the movie is listed as editable. Also, we set the `state` property of the looping menu items so that a check mark is displayed in the currently-active looping state item.

Handling the selection of one of our custom menu items is even easier than adjusting the menu items. We've already seen that we need to use the `call method` command to handle the "Select None" item. We can handle the looping menu items with pure AppleScript, as shown in **Listing 15**.

Listing 15: Handling menu item selections

```
on choose menu item theObject
    set theWindow to front window
    set theMovieView to movie view "movieView" of theWindow

    if name of theObject is "selectNone" then
        call method "selectNone:" with parameter theMovieView
    else if name of theObject is "noLooping" then
        set loop mode of theMovieView to normal playback
    else if name of theObject is "normalLooping" then
```



Getting rid of an old Mac?

Data on your Mac is recoverable even if you delete files, trash files, or reformat your hard drive. Permanently remove data with SuperScrubber's military-strength disk sanitization.

Erase the Risk for \$29.99



SuperScrubber

www.jiiva.com

© 2003 Jiiva, Inc., and its licensors. All rights reserved.



Application Builder Collection™

Reusable Cocoa frameworks for building solid Mac OS X applications

Helps you create professional-looking applications.

No deployment licenses or royalties.

Collection includes Rotational Slider, Formatters, Calendar View & Popup, and more.

 Jiiva

```

set loop mode of theMovieView to looping playback
else if name of theObject is "palindromeLooping" then
    set loop mode of theMovieView to ~
        looping back and forth playback
end if
end choose menu item

```

Closing a Movie Window

When the user quits ScripTeez, the movie window will close automatically. In an ideal world, we would first look to see whether the movie in the window had been edited and then prompt the user to save or discard any changes. To my knowledge, however, AppleScript does not provide any easy way to update the movie data in a movie file. So we'd need to use the **call method** command once again to call out to Objective-C methods. I'll leave that as an exercise for the interested reader.

ScripTeez does not provide any way to open a new movie window if we happen to close the movie window that's opened at application launch time. Accordingly, we should probably set things up so that closing the movie window will cause ScripTeez to exit. **Listing 16** shows the will-close method of the movie window.

Listing 16: Closing a movie window

```

on will close theObject
    quit
end will close

```

CONCLUSION

AppleScript's English-like language makes our code extremely easy to read; I doubt that anyone would have too much trouble understanding even the most complicated scripts we've encountered in this article. We can use AppleScript Studio's built-in commands and properties to handle a good deal of what's required to open and display QuickTime movies. Moreover, if need be, we can supplement our AppleScript with direct calls to Objective-C code to manipulate the Cocoa classes underlying our AppleScript Studio applications. This high-level scriptability and support for low-level method calling make AppleScript Studio an interesting addition to our QuickTime programming toolbox.

MacTech®
M A G A Z I N E

Get MacTech delivered to your door at a price **FAR BELOW** the newsstand price. And, it's **RISK FREE!**

Subscribe Today!
www.mactech.com

SOFTWARE LOCALIZATION MADE *easy*

POWERGLOT FEATURE HIGHLIGHTS

- LOCALIZE CLASSIC, CARBON™, COCOA® AND PALM OS™ APPS
- LEVERAGE EXISTING TRANSLATIONS
- AUTOMATE WITH APPLESCRIPT®
- IMPORT /EXPORT TRANSLATION MEMORIES

www.powerglot.com
browse, translate, click, done.

PowerGlot Software • Localization tools for Mac® OS
www.powerglot.com
info@powerglot.com

Mac OS, Carbon, Cocoa and AppleScript are trademarks of Apple Computer, Inc.
Palm OS is a trademark of Palm, Inc.

By Miro Jurisic

MindVision FileStorm

Creating Mac OS X installers with FileStorm

INTRODUCTION

The way users install your products often makes the difference between a happy user and a quick trip to the trash. Users know that confusion during the installation process is often a prelude to a baffling product. To make the best first impression, therefore, you need to understand the currently available software delivery tools on Mac OS X, and balance your own requirements and those of your users.

The range of possibilities is vast — from simple disk images and archives to complex interactive installers. This article reviews FileStorm by MindVision, a Mac OS X-only software delivery tool which allows you to create Mac OS 9 and Mac OS X disk images and Mac OS X installers.

OVERVIEW OF FILESTORM

FileStorm comes in two flavors, FileStorm and FileStorm Pro. FileStorm Pro is more expensive and offers more features. Both are royalty-free: you can distribute FileStorm installers and disk images without having to pay MindVision.

The main difference between FileStorm and FileStorm Pro is that FileStorm can only create disk images, whereas FileStorm Pro can also create Mac OS X installers. As of this writing, the price of FileStorm is \$19.95, and of FileStorm Pro \$79.95.

DRAG-AND-DROP INSTALLS

Apple's recommendation for all Mac OS X software is to use a drag-and-drop install if at all possible. This means that your users should be given a single item which they can install by dragging it into a location of their choosing in the Finder.

This method of distribution makes sense primarily for software which doesn't need to be in a special location in order to run, such as most applications. In contrast, a preference pane needs to be in one of `/Library/PreferencePanes` or `~/Library/PreferencePanes` in order to work correctly; consequently, authors of preference panes typically have to provide either an installer, or a document which explains to the user how to install the preference pane.

Even for software which can be installed with a simple drag, you still have to overcome the hurdle of actually getting the software on the user's computer. If the user can download your software over the internet, you typically have to encode the software in some way. If you use resource forks (without which you cannot deliver software that works on both Mac OS 9 and Mac OS X), you have to use an encoding which preserves the resource forks. If you don't use resource forks, your software almost certainly contains a Mac OS X bundle, which consists of several files and folders, and therefore needs to be archived in a single file from which it will be extracted on the user's computer.

An increasingly popular way of delivering files over the internet is Disk Copy disk images. Using a disk image allows you to encode an arbitrary hierarchy of files and folders into a single file. Their main advantage over other archive formats (such as StuffIt or gzip) is they integrate very well with the Finder: the user can view the complete hierarchy without going through a (possibly lengthy) decompression step, and you have full control over the visual layout of the hierarchy.

CREATING DISK IMAGES WITH FILESTORM

FileStorm offers a very elegant user interface for creating disk images, and also offers access to advanced features of disk images, many of which are not available in Apple's Disk Copy.

When you first launch FileStorm, a new project is created for you. A project consists of a disk image, which initially contains nothing but a custom background:

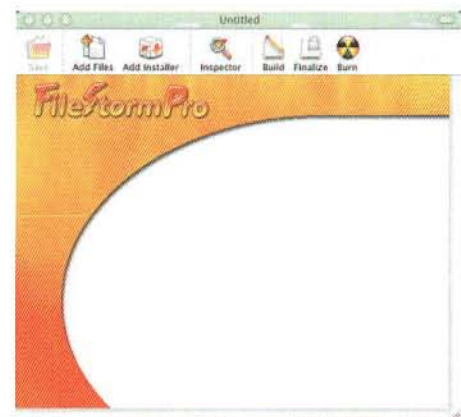


Figure 1. A new FileStorm project

Miro Jurisic is a wacky European working for Avid Technology. In his copious free time, he answers questions on `comp.sys.mac.programmer.*`, and thinks about chemistry. Visit him at <http://meeroh.org/>.

Manipulating the contents of a disk image is straightforward: drag files and folders in from the Finder, or add them using the Add Files menu command or the toolbar button. Once files and folders are in the disk image, you can rearrange them in the FileStorm window.

If you need precise control over the files' locations, bring up the Inspector window from the Project menu to set items' positions with pixel accuracy.

Unfortunately, FileStorm offers no advanced alignment and positioning tools. Its Align Horizontal and Align Vertical commands are very simplistic, and will suffice only for the simplest disk images.

Once you arrange the files, select Finalize Disk Image from the Project menu, and FileStorm will create a disk image that looks precisely as it did in your FileStorm project. You can mount this disk image and inspect it; by default, it is created in your Documents folder.

If you need to alter the disk image in a way that is not directly available in FileStorm, use the Build Disk Image command. This command creates a modifiable disk image for you to alter. After you are done with your changes, return to FileStorm and use the Finalize Disk Image command to build the final disk image.

Before you send the disk image to your users, you will probably want to change some of its settings from the defaults. To edit properties of the disk image itself, make sure that no items are selected by clicking on the disk image background, and bring up the Inspector window.

In the inspector, you can set the precise position and size of the main disk image window, as well as lock it against future modifications; this allows you to drag the FileStorm window around while you are working, without affecting the final location of the main disk image window on a user's screen.

The General category of disk image properties lets you edit the name and the icon for the volume; when the disk image is mounted, it will appear on the desktop with that name and icon. You can also configure the location at which FileStorm will put the disk image when it creates it, and add a background picture to your disk image. According to MindVision, FileStorm is currently the only tool that can produce a disk image with a background picture compatible with Mac OS X 10.1 and 10.2; disk images produced by other tools (including Disk Copy) fail to show the background image on either 10.1 or 10.2.

The background image is separate from the FileStorm watermark, which shows up in the top left corner of your FileStorm project window; the purpose of the watermark is merely to let you easily distinguish between your FileStorm project window and the disk image you may have opened in the Finder. The watermark never appears in the final disk images, and it can be hidden using the View menu.



Figure 2. A finalized disk image, mounted in the Finder



Now serving Cocoa[®] just the way you want it.

Training for Mac OS X doesn't have to be the same old flavor. Reserve your seat in a class at our scenic lodge location, or have experts come to you for **Extreme Mentoring**. Two weeks of on-site instruction and collaboration, customized to the requirements of your project. Book now for 2003. See why we're different.



**Big
nerd
ranch**

Intensive Classes for Programmers
www.bignerdranch.com

The Format category of disk image properties lets you choose exactly what type of file will be created by FileStorm. FileStorm can create disk images compatible with Mac OS 9, but they require a resource fork so need to be additionally encoded if you intend to deliver them on the Internet. Encryption and compression are also available, as well as several popular encoding format: BinHex, MacBinary, gzip, and bzip — the last two being only useful on Mac OS X.

Your disk image can be set to open automatically after mounting, or to automatically extract its contents and move itself to the trash instead of mounting. However, not all versions of Mac OS X work equally well with those two options; consult FileStorm help before you decide to use them.

You can set the size of your disk image. If you don't need to make any modifications to it when you finalize it, you should let FileStorm choose the size automatically. Otherwise, set the size to either a fixed amount, or a fixed amount beyond the amount needed by the items FileStorm puts on the disk image.

Finally, if you want to display a license agreement to your users, add your license agreement to the list in the License Agreements section of the disk image properties; the user will have to accept the agreement before the image can be mounted. You can provide several license agreements in different languages, and the one shown will be determined by the user's International preferences.

Once you are done with your disk image, FileStorm can burn it to a CD; your download and your physical media will thus have exactly the same look, pixel-for-pixel.

Anyone who has ever created a disk image knows that it involves a lot of fiddling — moving files around in the Finder, setting their positions *just so...* when you need to update a disk image with FileStorm, all you need to do is open the FileStorm project and click on the Finalize button. FileStorm will automatically grab the updated items off your disk and build an updated disk image with the files in the exact positions in which your FileStorm project puts them.

CREATING INSTALLERS WITH FILESTORM

While FileStorm excels at creating high-quality disk images, its power doesn't end there. Following the lead of Apple's PackageMaker, FileStorm Pro gives you the ability to create Mac OS X installers with a simple, straightforward user interface, similar to that of Apple's own installers.

As you probably know if you've installed any of Apple's software, Apple installers guide the user through a series of steps, including showing a license agreement and introductory documentation, selecting of a destination volume, and actually installing the software.

FileStorm Pro allows all of the same steps as Apple's PackageMaker, as well as additional ones: you can choose to show the user a graphic, an HTML file, or a QuickTime or an AVI movie. Furthermore, FileStorm Pro allows you to arrange all

those steps in whatever order you desire. Unlike Apple's Installers, which only allow you to present a set of three documents in a predefined order, FileStorm leaves you in control of the number and the order of the documents. For example, you can create an installer which shows one welcome document, installs some files, then presents a license agreement, and then installs more files.

To create a new installer in FileStorm Pro, you use the Add Installer command in the Project menu. The installer is added to the disk image, and you can set its properties by double-clicking it to bring up the Inspector window.

The General installer settings control some overall behavior of the installer, such as whether the installer needs the user to select a destination volume, or whether the installer needs administrative privileges to run. You can give your installer a custom icon and have it require a version of Mac OS X.

If your installer is so complex that users might need an uninstaller, you can have the installer automatically create an uninstaller, and put it in a location of your choosing.

The main part of the installer is in its Actions. An installer action can install an item on the user's system (at an absolute path, relative to user's home, or relative to the install volume), or it may present the user with some information; installer actions can be UNIX scripts, which give you considerable power over in your installer. The main thing lacking from installer actions is the ability to affect the flow of the installer itself; FileStorm Pro does not offer any ability for one installer item to influence another. That means that the user has essentially no control over the flow of the installer; the flow is entirely determined by you at the time the installer is created. As a result, just as in Apple's installers, you cannot have any optional install items; an item is either installed or it isn't.

Setting up an installer is as simple as dragging all the installer files into the Actions list in the Inspector window (or adding them using the Add Files command). Once they are in the list, you can arrange them to taste and set the action for each file. Initially, the action of each file is Install Item, which means that it will be installed on the user's system.

For any files which you do not wish to install on the user's system, you need to set the action to an appropriate value, such as Display Graphic (which displays a graphic to the user) or Support Item (which is used for files which are needed by other parts of the installer — for example, any images used by your Display HTML actions).

When you run a FileStorm Pro installer, its behavior is similar to that of Apple installers. The left side of the window has a buletted outline of the installer process, and the content of the current installation step is on the right; you can assign each installation step a custom title which appears in the installer outline.

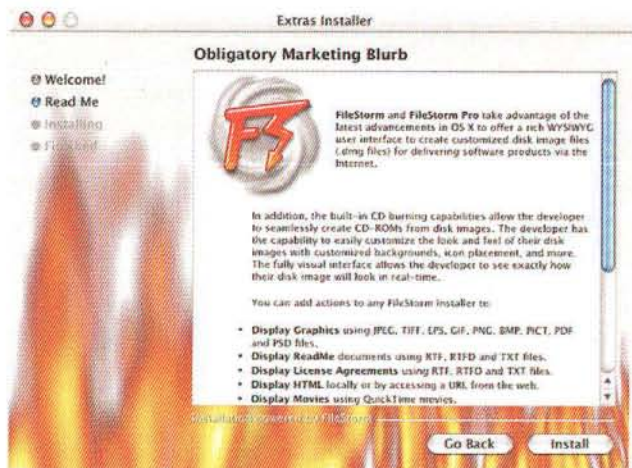


Figure 3. A FileStorm installer in action

If your installer needs to do something beyond the things directly provided for by FileStorm Pro actions, you have the full power of UNIX shell scripting available to you via the Execute Script install actions. Example scripts which come with FileStorm include changing ownership of permissions on installed files, opening files and folders, and adding login items to user's preferences. Since shell scripts can compile AppleScripts on the fly and execute them, you also have direct access to AppleScript from your FileStorm Pro installers; this is, indeed, how the Login Item shell script works.

Finally, in addition to all the shell scripts you can execute during the installation process, you can have a script execute immediately after the installer launches, or immediately before the installer quits, using the Advanced installer settings. If your installer produces an uninstaller, you can add those two options to your uninstaller as well.

INTEGRATING FILESTORM WITH YOUR BUILD ENVIRONMENT

FileStorm is fully scriptable using AppleScript, which allows you to control it from any build system which gives you the ability to execute arbitrary AppleScripts or shell scripts. Both Project Builder and CodeWarrior have this capability.

FileStorm comes with a number of sample AppleScripts; they show you how to control every aspect of project building, from adding files and folders to a disk image, to finalizing images and manipulating installers.

The main drawback of driving FileStorm via AppleScript is that it requires you to launch the FileStorm graphical interface in order to build a disk image. As a result, you cannot build a disk image with FileStorm unless you are physically logged in at the computer; doing it remotely via SSH doesn't work.

Most paths in the installer settings are treated as project-relative if they refer to files which reside in the same folder as the installer project itself, or a subfolder of that folder. This allows a project to be moved from one computer to another.

PhoneValet

Your Personal Telephone Assistant



- *Announces who is calling before you take the call.*
- *Reports to you how much time you spent on the phone and with whom.*
- *Locates the number in your phone book then dials it for you.*
- *Acts on your behalf when calls come in; selectively sends e-mails or invokes scripts.*

Talking Caller ID, Call Logging, Call-Triggered Actions and Voice Dialing on one or more lines.



Hardware, software and cables included!

www.phonevalet.com

See us at WWDC and Macworld Expo!

Developers and OEMs

Our hardware, drivers and APIs can be licensed to help you build exciting telephony solutions!

Parliant's USB Telephony Interface provides developers with the ability to dial the phone, decode Caller ID, detect dialed digits, play and record audio on and off the phone line, and get hook status.

Parliant offers high-level interfaces for Cocoa, Carbon, and C/C++ programmers to leverage our hardware without low-level USB programming.



Parliant Corporation

www.parliant.com

1-866-864-2334



Unfortunately, the location of the output image is not treated in the same manner, and as a result you have to be careful to use an output location which exists on every system you plan to build the installer on; otherwise, when you move the project to a new computer, you may get errors if the destination path doesn't exist.

The following shell script builds a FileStorm installer from CodeWarrior:

```
#!/bin/sh

# Copy the application into the installer input folder
ditto -rsrcFork "${MW_OUTPUT_DIRECTORY}/SurfWriter.app"
"${MW_OUTPUT_DIRECTORY}/Installer/Binaries/SurfWriter.app"
# Build the installer using AppleScript via osascript
# Note the try/quit/repeat loop -- FileStorm returns from the
# finalize event before the image is finalized
osascript -s o << EOF
with timeout of 3600 seconds
    tell application "FileStorm"
        activate
        open
        (($MW_OUTPUT_DIRECTORY)/Installer/SurfWriter.fsproj" as
        POSIX file) as alias)
        tell first document
            build image with replacing
            finalize image with rebuilding
        end tell
        repeat while true
            try
                quit
            exit repeat
            on error
                delay 5
            end try
        end repeat
    end tell
end timeout
EOF
```

SUMMARY OF FILESTORM CAPABILITIES

If you want a simple drag-and-drop installation, FileStorm gives you a great way to deliver your software with minimal hassle and a high-quality presentation. By using the option to automatically replace the disk image with its contents (only available on Mac OS X 10.2.3 and later), you get a self-extracting package; the only thing the user has to do after downloading it is double-click.

If you have a more complex software package consisting of a multitude of items, but do not require an installer, FileStorm lets you create a disk image with your software; you get a high degree of control over the layout, appearance, and behavior of the disk image, and you can burn the software to a CD with a click of the mouse.

If you need an installer for your software, FileStorm Pro offers a great deal of flexibility in customizing your installer, while retaining the simple and straightforward user interface pioneered by Apple's installers. FileStorm Pro installers are not without limitations; a big drawback for large software installations is the lack of control over installer flow.

However, if FileStorm Pro installers satisfy your needs, using a FileStorm Pro installer on a self-extracting disk image is a marvelously simple way to deliver an excellent installer with minimal hassle for both you and your users.

CONCLUSION

FileStorm and FileStorm Pro are both excellent products and priced very fairly. The user interface is superb, and has the polish of a well-designed Mac application — one that makes you feel in control of your project and helps you get your work done.

No utility provides a user interface for creating disk images that is cleaner or more powerful than FileStorm's. By using FileStorm to create disk images, you can save yourself a great deal of tedious fiddling in the Finder, as well as eliminate a bunch of specialized AppleScripts and shell scripts from your workflow.

The user interface of installers produced by FileStorm Pro is elegant and familiar, and they provide flexibility significantly beyond that of Apple's PackageMaker. There are more types of install actions and their order and numbers can be varied much more so than with PackageMaker.

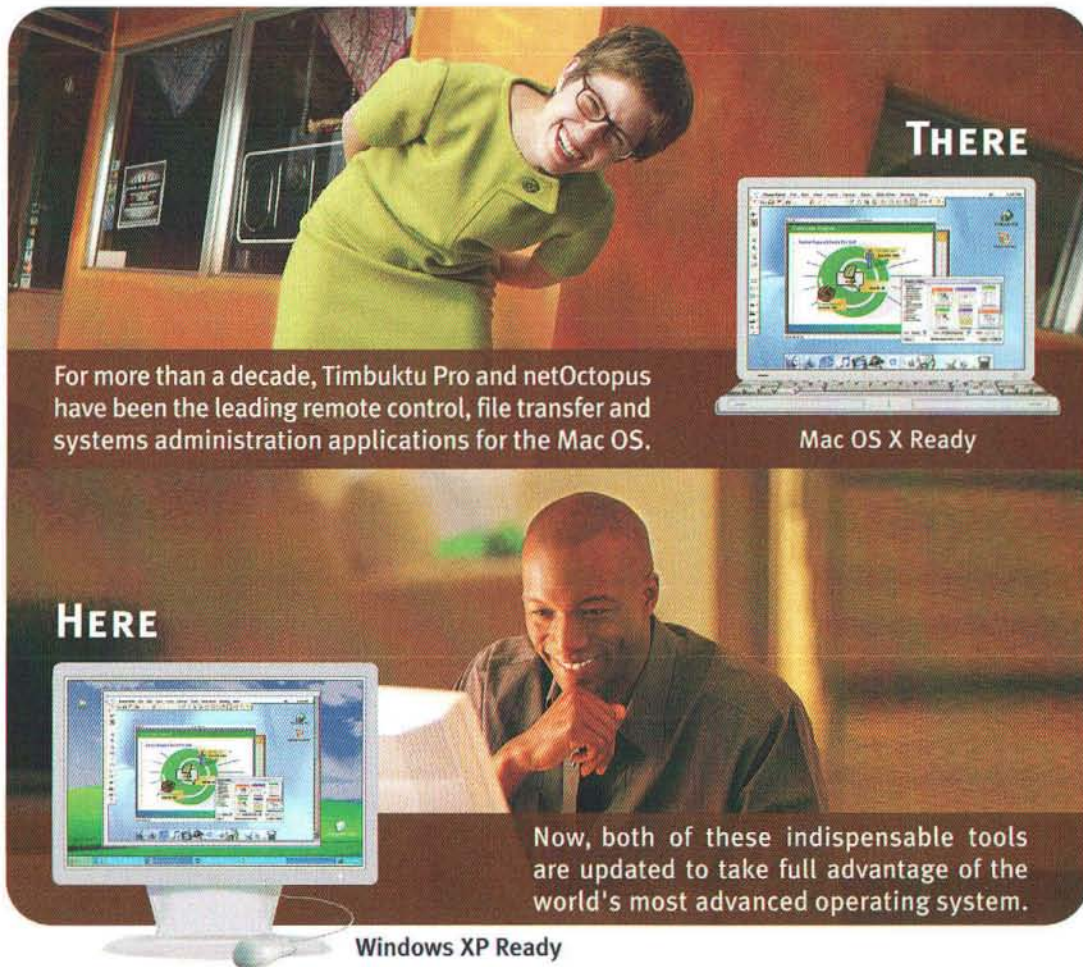
Two features of PackageMaker missing from FileStorm are the ability to create installers without being logged in at the computer, and the ability to create subinstallers (metapackages, in PackageMaker parlance).

Unfortunately, there is no way to dynamically control the flow of a FileStorm Pro installer at runtime. If you need the ability to conditionally install an item based on the state of user's system or input from the user, you won't get very far with FileStorm. MindVision's current recommendation is that you get a copy of Installer VISE, their flagship installer product for Mac OS 9 and Mac OS X.

However, once you've used FileStorm, using Installer VISE is a significant setback in elegance and ease of use; installers generated by Installer VISE are less straightforward less coherent than the ones generated by FileStorm. I can only hope that using Installer VISE is a temporary workaround, and that MindVision will enhance FileStorm so that it can more effectively compete with Installer VISE on Mac OS X — both because FileStorm is much more pleasant to use, and because it makes my users' lives better.

FileStorm Pro installers require Mac OS X, so if you need a Mac OS 9 installer, you have to use a different product (possible Installer VISE); however, if you only need a disk image to deliver your software, FileStorm does allow you to create ones that work on Mac OS 9 and Mac OS X.

Stay In Control Wherever You Go.



THERE

For more than a decade, Timbuku Pro and netOctopus have been the leading remote control, file transfer and systems administration applications for the Mac OS.

Mac OS X Ready

HERE

Now, both of these indispensable tools are updated to take full advantage of the world's most advanced operating system.

Windows XP Ready

Timbuku Pro

Whether you're at home or at work, Timbuku Pro allows you to operate distant computers as if you were sitting in front of them, transfer files or folders quickly and easily, and communicate by instant message, text chat, or voice intercom.

<http://www.timbuktopro.com>

netOctopus

Intuitive and powerful, netOctopus can manage a network of ten or 10,000 computers. Inventory computers, software and devices on your network; distribute software; configure remote computers; and create custom reports on the fly.

<http://www.netoctopus.com>

Learn more, try it, or buy it online. Call us at **1-800-485-5741**.



timbuku® • netOctopus®

netopia.

By Kevin Hemenway, *Questionable Intender*

User, Meet Apache. Apache, Hug.

Or: How To Learn More About Your Cuddly New Web Server.

Hopefully, if you were at all interested in the last column (*MacTech*, June 2003), you've mentally prepared yourself for the coming months of web serving hemming and hawing. I'll continue to assume that you know more about your own network and internet provider than I ever could, and as such, will only touch briefly on ISP-related workarounds. In this installment, we'll turn on our web server, explore its directory structure, and learn how to interact via the command line. You should know how to operate Apple's Terminal application – if you don't, I highly recommend brushing up on Chris Stone's series, *Learning the Terminal in Jaguar*, on MacDevCenter.com.

HARTMAN: TODAY YOU PEOPLE ARE NO LONGER MAGGOTS

A word of encouragement: 63% of the web sites on the Internet are using Apache (according to a 2002 report from <http://netcraft.com/>). Apple, convincingly enough, has included Apache in its own OS X and OS X Server products. It isn't a "ported" version or a slimmed-down feature set, but rather a full-fledged implementation of Apache with all the fixin's. Much of what you learn in these articles will apply to any installation of Apache, regardless of whether it's on Mac, Linux, or even Windows.

If you're familiar with the differences between Mac OS X and Mac OS X Server, you're probably aware that Server contains more GUI based administration tools for programs like Apache, sendmail or MySQL. This doesn't indicate that one Apache is better than the other – everything you can do in Server can be done in the consumer OS, and vice versa. As of this writing, these articles assume you're using the consumer Jaguar, and not the Server version (which we won't be covering).

Alternatively, a word of possible disillusionment: much like any web developer worth his salt will use a raw text editor like BBEdit (which I prefer, see <http://www.barebones.com/>) to code their HTML, most system administrators eschew the need for hand-holding GUI tools and dive right into text editing a configuration file. This is decidedly un-Mac-like but, until a few years ago, so was the thought of including Linux (or, more accurately, BSD) as an under layer. If you want to become proficient at web serving or programming, you're going to have to get used to editing configuration files. Yes, there are GUI based tools available, but you'll do a lot better if you learn to fish, rather than plunking quarters into a vending machine (that made sense, right? Right!)

Enough soap-boxing. Commence the casting.

KAHN: BUTTERFLY IN THE SKY... I CAN FLY TWICE AS HIGH!

There are a few different ways you can get a rise out of Apache, the most immediate of which is through the Macintosh GUI. In Jaguar, this setting is hidden underneath the *Sharing* System Preference; open that now (*Apple Menu > System Preferences... > Sharing*). The first tab we see, *Services*, lists a number of capabilities we can turn on or off, as well as our current network address (which may or may not be accessible to the outside world).

In **Figure 1**, you'll see that "Personal Web Sharing" has been highlighted. To turn our Apache web server on, either put a check in the box on the left, or click the "Start" button on the right. A few seconds later, we'll see the results shown in **Figure 2**. Of special interest is the new information at the bottom of the screen – the first URL is the home of the primary web site of your machine, and the second is the address of the current user's personal web site.

Kevin Hemenway, coauthor of *Mac OS X Hacks*, is better known as Morbus Iff, the creator of disobey.com, which bills itself as "content for the discontented." Publisher and developer of more home cooking than you could ever imagine (like the popular open-sourced aggregator AmphetamineDesk, the best-kept gaming secret Gamegrene.com, articles for Apple's Internet Developer and the O'Reilly Network, etc.), he's an ardent supporter of writing incorrect passwords on sticky notes, just to confuse peepers. Contact him at morbus@disobey.com.

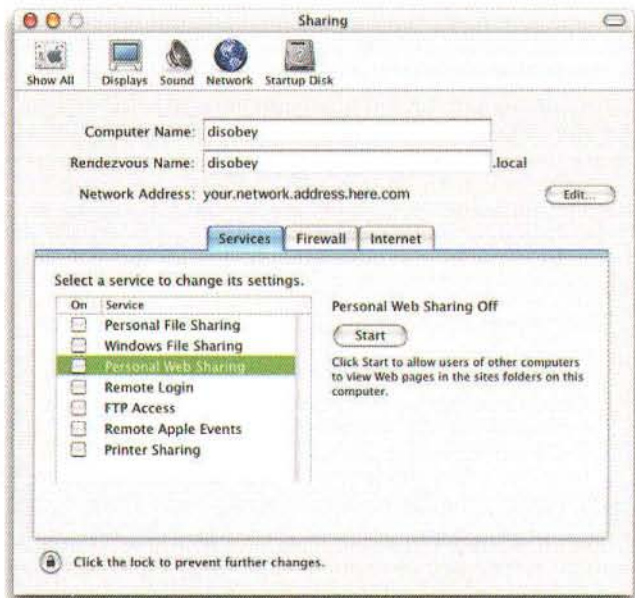


Figure 1. The Services tab of the Sharing System Preference.

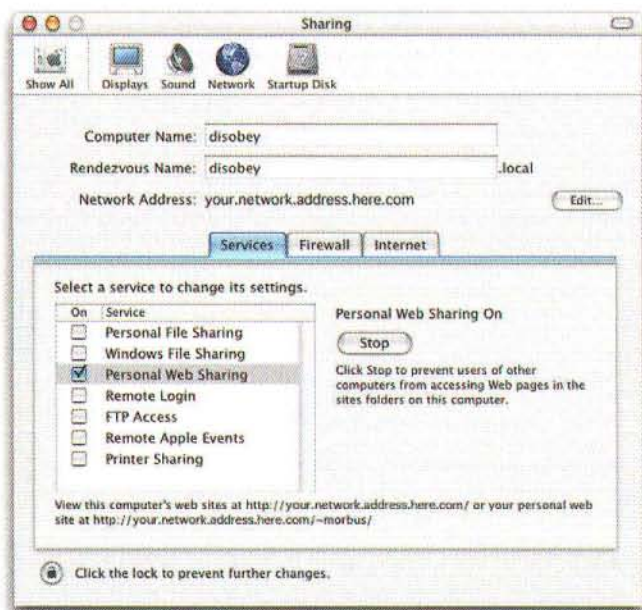


Figure 2. The Apache web server has been started.

Depending on your network or ISP's configuration, you may be able to type (or cut and paste) those URLs into your browser's address bar and see the default pages of your built-in web sites (Figure 3 and Figure 4). If you don't see those pages, or else get an error message concerning connectivity, you should run through some of the steps in last month's column to try and determine your external IP and whether it's

data
There are Users
data
and Losers...

Which
would you rather
be

Introducing
Data Safety System
Backup, Undelete and Recover files.

DATA
Backup

DATA
Recycler

DATA
Rescue

Data Users get it!

www.prosofteng.com
PROSOFT
engineering, inc.

These products are only available for the Mac OS, so your Windows friends will still be losers...

©2003 Prosoft Engineering, Inc., All rights reserved

viewable to the outside world. For now, you should be able to follow along by using `http://127.0.0.1/` and `http://127.0.0.1/~username/` respectively.

Figure 3 shows the default web page that is shipped with most Apache distributions – it's just a quick confirmation that the Apache web server is up and running smoothly. Ultimately, you should see this page only once (that once is now). You'll also be informed that the Apache documentation can be accessed from `http://127.0.0.1/manual/`.

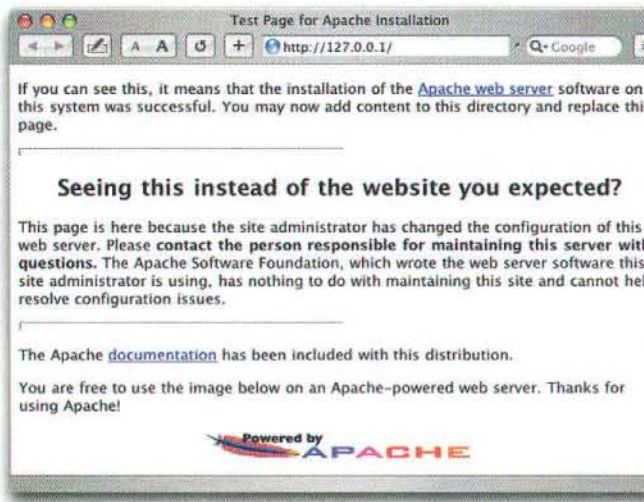


Figure 3. The default root web page of Apache.

Figure 4, on the other hand, shows a friendly blurb that Apple created to ease new users into their virginal web serving experience. It briefly covers what we just did (turning on the web server) as well as quick definitions of HTML and Apache. You should read over both the default pages – you'll probably be deleting the files that represent them shortly.

But, where are these files physically located? The first URL, being the root location of the web server, is served, semantically enough, from the root web directory of Apache: `/Library/WebServer/Documents`. The second URL, being user specific, is served from `/Users/username/Sites` (as explained in **Figure 4**).

And that, as they say, is that. Your Apache web server is running, you're seeing the default web pages, and you could publish vanilla HTML with nary a peek at the underlying sprockets and cogs. So far, though, this hasn't been very satisfying. It's hard to feel good about yourself when all you did was click a button and piddly-type a URL.

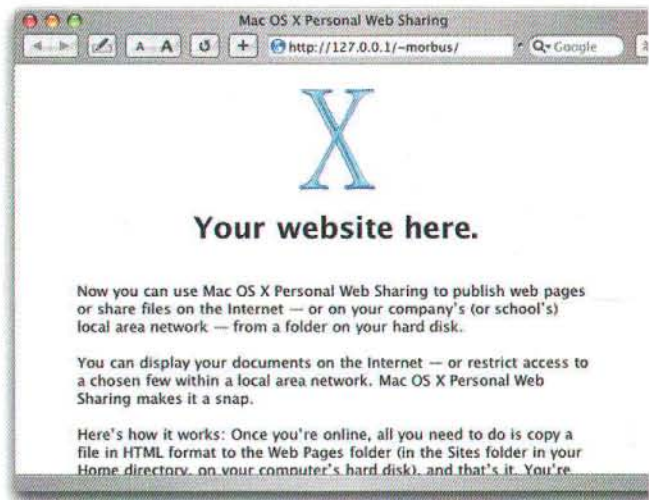


Figure 4. The default user web page from Apple.

BATISE (TRANSLATING): IF NO PAIN, NOTHING GOOD IS BORN

Being spelunkers of the technical sort, however, it's time we dig deeper. To do so, we'll open up a Terminal and type `httpd -V` which gives us the results shown in **Figure 5**. What we're doing is asking Apache (represented by the shell program `httpd`) to show its compile time settings. These will change depending on your distribution (Redhat's output will be different from SuSE's output, which will be different from OS X's, and so forth). I'll explain some of the more important entries below.

Note: The screenshot, and my explanations, are based off version 10.2.6 of OS X. If you're following along with 10.1 (waiting for Panther, eh?) or earlier versions of 10.2, you may see a slightly different output. Don't fret... the differences rarely equate to much importance, and when they do, I'll make a glib comment.

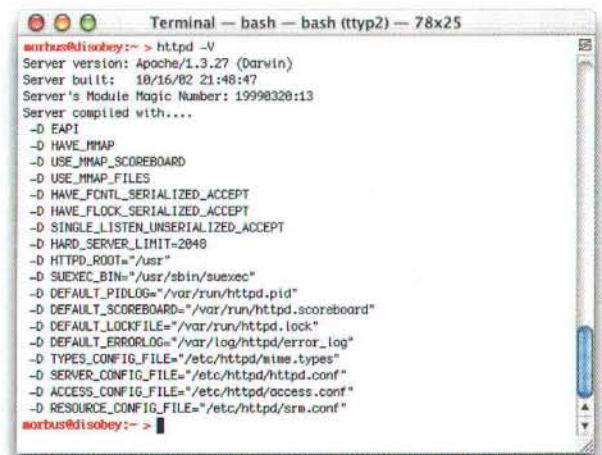


Figure 5. The results of an `httpd -V` shell command.

TECHNICAL DIAGRAMS AND REPORTS

PLOTS • GRAPHS • CHARTS • VISUALIZATION
TEXT • TABLES • FORMS



CONTRACTING

SPECIALIZING IN BUSINESS AND TECHNICAL REPORTING
DATA FROM ANY SOURCE

WHY SETTLE FOR LESS WHEN YOU CAN HAVE THE TOP-MOST EXPERTS?

PROVIDING BUSINESS, ENGINEERING AND SCIENTIFIC REPORT SYSTEMS SINCE 1989

Vwidget Pro™:

A programming kit to control and report data in visual formats.

Peer Visual™:

A server to distribute dynamic visuals to the web and printers.

Vwidget Fusion™:

A distributed and parallel network data acquisition kit.

Vwidget Signals™:

A real-time multi-channel strip chart and signal display kit.

Vwidget User™:

A graph building tool, palette, and preconfigured graphing server.

Limited Time Offer: \$89 (\$49 for students)(this ad made with Vwidget User)

- Technical Analysis
- Real-Time Trading
- Laboratory Instruments
- Process Control
- Database Interfaces
- Modeling
- Device Drivers
- Data Archival
- Mac® OS X Experts

VVI®

www.vvi.com

info@vvi.com

888-VVI-PLOT

The first bit of info is the server version of the currently installed Apache (and, for the esoteric, the timestamp of when it was actually compiled). Thankfully, Apple has generally been responsive with their security updates, and most OS X users are running the latest release (1.3.27, although, at the time of this writing, there were rumors of an impending 1.3.28).

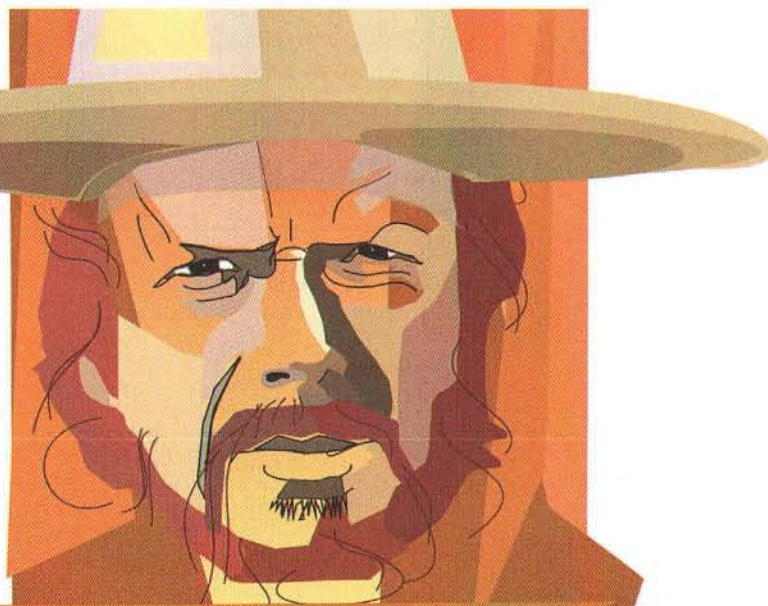
The next line worth exploring is `-D HTTPD_ROOT`, which tells us where the Apache binaries have been installed. The most important files, like `httpd` and `apachectl`, live in `/usr/sbin`. All of Apache's modules (read: plugins) live in `/usr/libexec/httpd`. If you've had experience with Linux programs before, this layout is relatively familiar. The next line, concerning `SUEXEC_BIN`, can safely be ignored – `suexec` isn't enabled or shipped under OS X (we can, however, recompile Apache and add it ourselves. Long story. Eventually.)

`-D DEFAULT_ERRORLOG`, the next entry of importance, is probably the single greatest answer to all your problems before, now, and after. Whenever something goes wrong, check your error log. Whenever something goes right that shouldn't have gone right, check your error log. Whenever you suspect someone is chuckling behind your back, the error log will have their home address. "Check your error log" is the Apache equivalent of "RTFM" – before in-the-know users will answer any of your tech support questions, they'll want to know what the error log says. More often than not, the error log will tell you exactly what went wrong. Don't be embarrassed. Check your error log. The quickest and most helpful way is with `tail /var/log/httpd/error_log` which spits the last ten lines of any file you pass to it.

We'll talk a bit more about log files in an upcoming column, but for now, realize that a matching `/var/log/httpd/access_log` covers successful operations (in the sense that the original URL request garnered a "proper" response). In previous versions of OS X, you would have seen a matching `-D DEFAULT_ACCESSLOG` in **Figure 5's** output. This has since moved inside the configuration file, which is what our remaining four lines cover.

The first of the last, `/etc/httpd/mime.types`, contains the mapping between a file extension and the MIME type sent to the browser (or, more generically, the "requesting user-agent"). For now, we'll leave the actual definition of MIME types to a later column, but if a `.jpg` were served as `text/html` (and not its proper `image/jpeg`), then the browser wouldn't be able to properly render and display the picture.

The next entry is The Big One – `/etc/httpd/httpd.conf` points to the file that handles all the configuration of our Apache web server. It's suitably large, suitably commented, and suitably intimidating, enough so that the included early warning should be taken to heart: *Do NOT simply read the instructions in here without understanding what they do. They're here only as hints or reminders. If you are unsure consult the online docs. You have been warned.* The last two entries in our output can be safely ignored – they're deprecated configuration files that have since been merged with the master configuration. If you get caught using them, your error log will spit out my home address.



"Well, what're you waitin' for? Draw!"

Eazy Draw

Make Drawing Fun on OS X

Introducing EazyDraw – the fun, easy-to-use Mac OS X design tool that lets you draw like a pro! Now you don't need to be a graphic artist to create great illustrations. EazyDraw's vector-based graphics and editing capabilities make it easy to create technical diagrams, flow charts, and business communications as well as commercial line art illustrations and graphic elements for application software and web design.

Learn more about EazyDraw today! Get big savings buying direct from our online store. Visit us at **www.eazydraw.com** (That's easy with a Z).

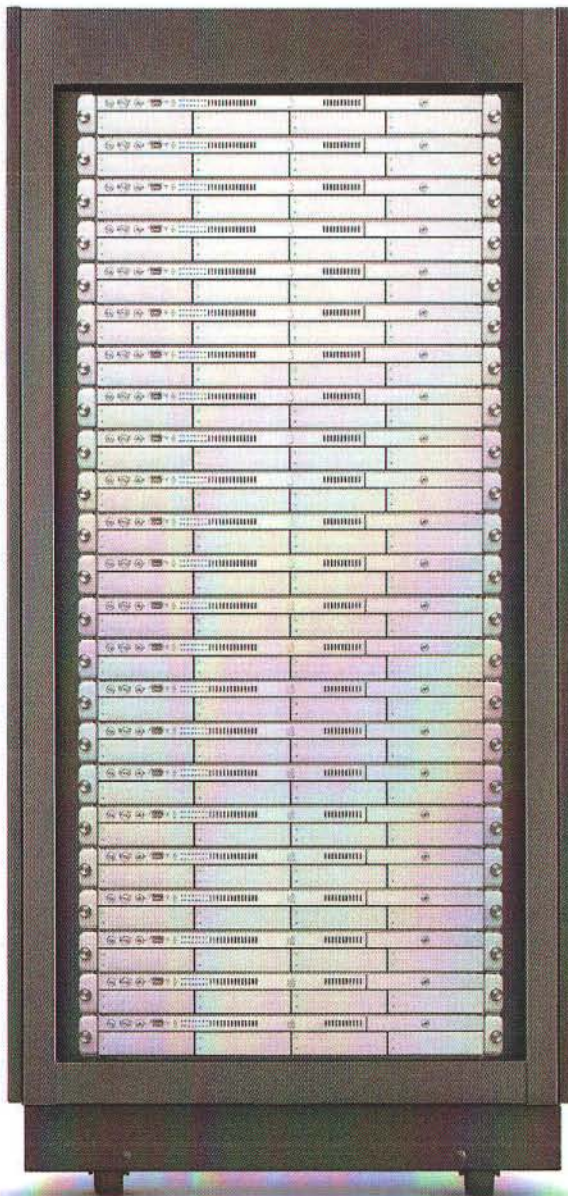
© 2003 Dekorra Optics, LLC. All rights reserved. EazyDraw and the 'box of tools' are trademarks of Dekorra Optics, LLC. Mac and Built for OS X are trademarks of Apple Computer, Inc.



Xserve



Density optimized rack mounted Mac OS X Server



UNIX-based Server Solutions from Apple
Nearly half a terabyte of storage per 1U
630 gigaflops of processing power
Hot-Swappable drives
Industry-standard 1U rack-optimized design

There has never been a better time to buy...
Small Dog Electronics carries
factory refurbished models too
offered at substantial savings!
(subject to availability)



**Small Dog
Electronics**
www.smalldog.com

1673 MAIN STREET, ROUTE 100, WAITSFIELD, VERMONT



Apple Specialist

To learn more: <http://www.smalldog.com/xserve/>

HAMMOND: ALL MAJOR THEME PARKS HAVE DELAYS

If you recall from the first column, one of the ways an ISP can put a dent in your web serving plans is by filtering incoming HTTP traffic. With such a filter, any incoming requests on port 80 will be dropped, and they'll never reach your anxious and ever-ready Apache. Working around this is easy enough to serve as an introduction to editing the master configuration file. **Note:** For those who DON'T have an evil ISP, just mentally follow along – you'll have less mundane things to do come next column.

The first hurdle concerns saving our upcoming changes to `/etc/httpd/httpd.conf`. This file, being "special", requires heightened privileges to allow modifications – privileges that your user account doesn't have by default. To get these special administrative privileges in the Terminal, you'll need to preface your intent with the `sudo` command. Said command gives you, for that one instant, super powers – enough so to save your changes to an otherwise protected file. The downside of using `sudo` is that you need to be proficient in a shell editor like `pico`, `emacs`, or `vi`.

Myself, I prefer BBEdit 7.0's shell utility (which can be installed via their *Preferences > Tools > Install "bbedit" Tool*). BBEdit's utility is smart enough to know that when you attempt to save a protected file, you should be prompted for an administrative password. In my case, I'll launch into our next paragraphs with `bbedit /etc/httpd/httpd.conf`. For those without BBEdit, utter `sudo EDITOR /etc/httpd/httpd.conf` where `EDITOR` is your preferred program of choice. If you're new to shell editing, there's a quick tutorial on using `pico` in Chris Stone's *MacDevCenter.com* series.

However you get there, we should now be looking at Apache's primary configuration file. Within the first screen of information, you should see the warning I italicized above, and I'll warn you again: here they be dragons. Friendly Puff-like dragons, but dragons nonetheless.

In general, when you want to modify the configuration of Apache, more often than not, just do a search for your desire and you'll find something worth investigating. In our example, we've got problems with the ISP filtering port 80 traffic, so do a search for the word "port". You will find, soon enough, the following bit of text:

```
#
# Port: The port to which the standalone server listens.
# For ports < 1023, you will need httpd to be run as root
# initially.
#
Port 80
```

Here, you can see that Apache has been configured to start up on the default (and expected) HTTP port 80. Since our theoretical ISP has blocked that, we need to change it to something else. Good alternative choices are 8000, 8080, or 8088. Sadly, whatever you choose will turn all your URLs ugly... if you

chose 8000, you'll be forced to give out `http://127.0.0.1:8000/~morbus/` instead of `http://127.0.0.1/~morbus/`. Choose your poison, and save the file.

Since we've made a change to the configuration, we've now got to restart Apache. You'll become familiar with these steps as the column progresses: any time a change is made to the configuration, it won't be put into play until Apache is stopped and then started. Before we actually do that, it's often handy to run `httpd -t` first. Much like `-V` gave information about the built-in compile time settings, `-t` gives us some insight on our configuration file by testing it for errors. If everything's grand, we'll get a **Syntax OK...** if not, we'll be told on what line we actually screwed up. The benefit of testing before restarting should be obvious – it gives us a chance to fix problems before our visitors start complaining.

We can restart Apache one of two ways: by toggling the buttons in our *Sharing System Preference* (refer back to **Figure 1** and **Figure 2**) or by using the `apachectl` shell utility. I personally prefer the shell utility, due to a shortcoming in the System Preference: if your `httpd.conf` has an error in it, the System Preference will attempt to "start" indefinitely, expecting a positive response that it will never get. With `apachectl restart` (or its implied brethren: `apachectl stop` and `apachectl start`) you'll be given an `httpd -t` diagnosis if anything goes wrong. Once Apache is restarted, the configuration change has taken effect, and you should be able to visit your newly tweaked `http://127.0.0.1:8000/~morbus/` (or whatever actual port you chose).

HOMEWORK MALIGNMENTS

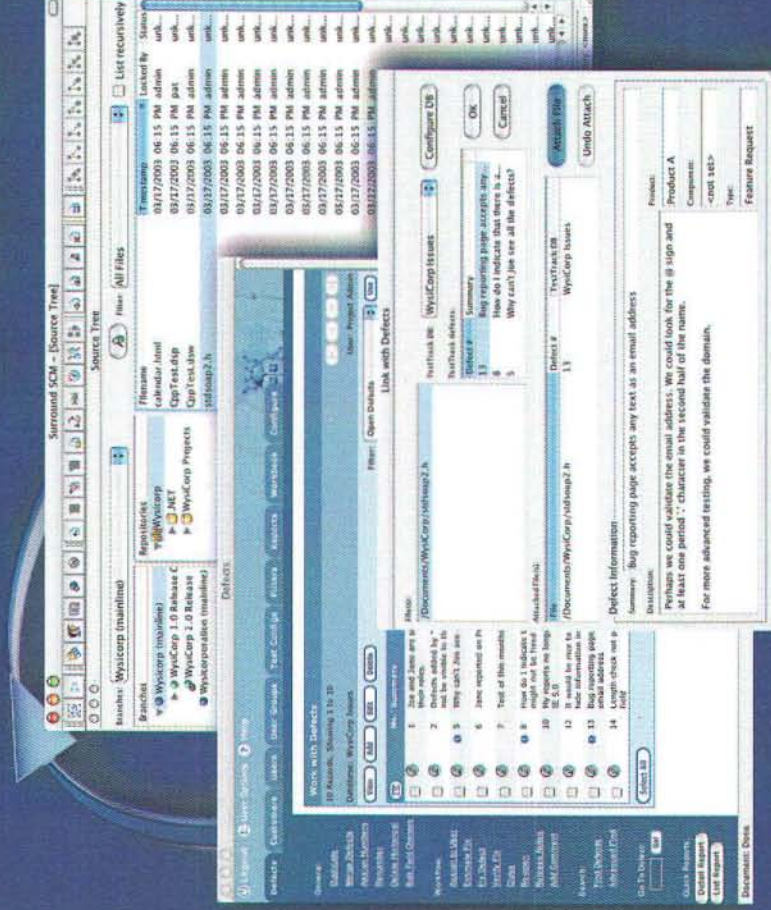
In our next column, we'll begin exploring our first major feature: server side includes. Often ignored for being "too simple", they can do some fairly useful things without much effort. Until then, students may contact the teacher at `morbus@disobey.com`.

- Peruse through the `/etc/httpd/httpd.conf`. Familiarize yourself.
- The Apache documentation already loaded on your machine (see **Figure 3**) is some of the best open source documentation around. I'll refer to it from time to time.
- Liked reading about URL design from the first column? Check out "Toward Next Generation URLs" by Thomas A. Powell and Joe Lima: <http://port80software.com/support/articles/nextgenerationurls>
- Complete the ever-enlarging animal tree: maggot, butterfly, what, and what?
- Each of the headings is a quote from a movie or TV show. Name them.

Complete Source Control and Defect Management

Seapine Software™
*changing the world
of software development*

for Mac OS X



Effective source code control and defect tracking require powerful, flexible, and easy-to-use tools—Surround SCM and TestTrack Pro

- Complete source code control with private workspaces, automatic merging, role-based security, and more
- Comprehensive defect management — track bug reports and change requests, define workflow, customize fields
- Fast and secure remote access to your source files and defects — work from anywhere
- Advanced branching simplifies managing multiple versions of your products
- Link code changes with defects and change requests — know who changed what, when, and why
- Scalable and reliable cross-platform, client/server solutions support Mac OS X, Windows, Linux, and Solaris
- Exchange data using XML and ODBC, extend and automate with SOAP support
- Licenses priced to fit your budget

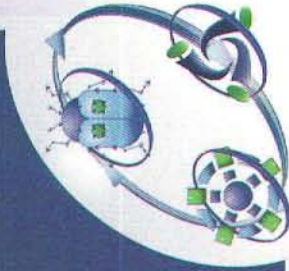
Seapine
Surround SCM
Seapine
TestTrack Pro



all product names listed herein are registered trademarks of their respective owners. All rights reserved.

Seapine Software Product Lifecycle Management
Award winning, easy-to-use software development tools

Download Surround SCM
and TestTrack Pro at
www.seapine.com
or call 1-888-683-6456



By Scott Knaster

Panther and G5: The Future is Almost Now

This month we take a detour from our usual dusty dirt road to examine, MacTech-style, Steve Jobs' keynote speech at Apple's Worldwide Developers Conference 2003 in San Francisco.

A SUNNY DAY AROUND THE BAY

This past spring, Apple announced that it had moved WWDC in time and space, relocating from its traditional May dates in San Jose to June in San Francisco. But the biggest change was the buzz that WWDC had stepped up in rank and was now to host an Apple product announcement of the highest order.

The conference was held at the wet-paint-new Moscone West venue. You can tell how modern this place is when you visit the bathrooms: there are motion-sensitive automatic paper towel dispensers. When I saw these, I wondered if I was witnessing one of Apple's big technological announcements of the day. But I wasn't – Apple hasn't gotten itself into sinktop computing just yet.

As with any Steve Jobs keynote, there was electricity in the air and the lines for the keynote hall formed early. Just before the scheduled 10:00 AM start, the doors opened and the attendees surged in. Apple's crowd-control folks were heard to ask people not to run. Everyone found a seat.

ANTICIPATION

As we waited for the show to begin, everygeek took out a PowerBook or iBook and began idly scanning the Internet. It soon became apparent that AirPort access wasn't working very well. So enterprising attendees starting setting up computer-to-computer networks and using the *names of the networks themselves* as a form of communication. Some of the first few networks to appear were named "airport not working again", "I love steve jobs", "I'm running a 970 pbook", and the clever and enticing "Steve Jobs' computer". By using the names of networks to converse, Mac developers in the crowd fulfilled the old saying that "Eventually, every application expands until it is able to send e-mail."

By using MacStumbler, an open source utility for sniffing out wireless networks, I was able to take a snapshot of this playful networking (see **Figure 1**).

SFO/Internet	02:00:B5:F8:8F:AA	1	24	Ad-hoc
iChat here	02:00:D1:09:4D:9A	3	39	Ad-hoc
AirPort_G_Level2	B2:F9:BB:AE:72:28	1	20	Ad-hoc
Don't Run -- Walk	02:00:C6:03:1D:D6	11	12	Ad-hoc
Here comes the dodecahedron	02:00:9C:B0:66:D9	11	25	Ad-hoc
Stop it...	02:00:B5:D4:DC:CF	11	15	Ad-hoc
SFO/Internet	02:00:24:80:1E:D2	1	20	Ad-hoc
threerings\enet	02:00:D9:24:D9:D4	11	20	Ad-hoc
AirPort_G_Level2	02:00:16:F8:10:23	1	13	Ad-hoc
AirPort_G_Level2	3A:53:72:0D:1B:FA	1	19	Ad-hoc
GRC Wireless	02:00:5E:EF:00:B6	1	21	Ad-hoc
Impeach Bush!	02:00:13:3C:72:0E	1	12	Ad-hoc
Yeah, I'm on a Quad 3GHz 970!	02:00:AB:D9:F1:C1	11	27	Ad-hoc
Where did all my networks go?	02:00:67:9E:00:70	11	16	Ad-hoc
All your Network are belong to Us	02:00:FF:DE:E9:07	11	60	Ad-hoc
Macteens forever.	02:00:43:C6:82:E5	11	36	Ad-hoc
BetterLeaveAirportChanFreePlz	02:00:6B:92:6E:5A	11	32	Ad-hoc
Anyone here want to cyber?	02:00:46:9F:C3:EF	11	19	Ad-hoc
Putah Creek Development	02:00:21:B4:A0:1F	1	24	Ad-hoc
home	02:00:A9:BE:6E:A1	1	29	Ad-hoc
now remember, no cameras	02:00:DA:A1:E6:24	11	18	Ad-hoc
macosxslabs	02:00:8C:6E:62:43	11	25	Ad-hoc
we are geeks tiking with essids	02:00:F8:D3:F8:23	11	24	Ad-hoc
MacMedics	02:00:B3:15:44:DC	1	27	Ad-hoc
SNAPPLE	02:00:5F:BA:DD:2B	1	25	Ad-hoc
Memo: stop whining! -Steve	02:00:92:68:17:18	11	27	Ad-hoc
Apple Store Position Opening	A6:E5:B6:C7:D7:D8	10	23	Ad-hoc
and to think i was going to work	02:00:08:24:08:D4	11	16	Ad-hoc
Steve Jobs' Computer	02:00:71:C9:9E:C9	11	27	Ad-hoc
heaven forbid we leak something	92:60:3F:13:1A:63	1	14	Ad-hoc
GAWD THIS IS PATHETIC	32:19:32:E9:CC:FC	11	17	Ad-hoc
I love Steve Jobs	02:00:DF:E9:D7:76	11	25	Ad-hoc
NO AIRPORT AGAIN??????	02:00:38:4C:11:5C	11	26	Ad-hoc
2GHz 970 Machine...Really	66:85:29:6D:1D:FB	1	18	Ad-hoc
ntwk running on mslt i guess	02:00:2E:C6:2E:36	11	19	Ad-hoc
I feel the RDF already!!!!	02:00:22:91:57:96	11	27	Ad-hoc
Log live iBook, they get connecte	02:00:AB:92:EB:34	11	26	Ad-hoc
Nope, Don't think so.	02:00:D1:0B:D9:94	11	31	Ad-hoc

Figure 1. With Internet access failing, clever geeks at the WWDC keynote expressed themselves via the names of private networks they created.

Scott Knaster has been writing about Macs for as long as there have been Macs. Scott's books *How To Write Macintosh Software* and *Macintosh Programming Secrets* were required reading for Mac programmers for more than a decade. Scott wrote developer books for General Magic and worked on Mac software for Microsoft. Scott's books have been translated into Japanese and Pascal. Scott has every issue of Mad magazine, which explains a lot.

Ask yourself this question...

I use my Mac for:

- ☐ Programming as a hobby
- ☐ Exploring the depths of Mac OS X
- ☐ QuickTime Development
- ☐ Application Development
- ☐ Network Administration
- ☐ All Of The Above

...this is the answer:

 **MacTech[®]**

The Journal of Macintosh Technology and Development

Whether you program as a hobby or are a full-time developer, MacTech gives you the info you need from the people that know.

www.mactech.com

PO Box 5200 • Westlake Village, CA • 91359-5200 • orders@mactech.com
Toll-Free: 877-MACTECH • Outside US/Canada: 805-494-9797 • Fax: 805-494-9798

Steve strode out without introduction (everybody knew who he was anyway) and commenced his presentation with a review of what's happened recently and how things are going. He announced impressive numbers for AirPort Extreme ("1000% sure" that 802.11g is the successor to 802.11b), Apple retail stores (58 stores opened in 25 months so far, with cool new giant stores that look like enormous iPod boxes coming to Chicago, San Francisco, and Tokyo), and iPod (1 million shipped and 5 million songs purchased at the iTunes Music Store). He announced that Safari 1.0 was now available, along with a framework that lets you build a browser core into your own apps with incredible ease.

Of course, nobody was there to hear that stuff: they wanted to find out what was new. Steve kicked off that portion of the speech by talking about what was going on with OS X. Again with the impressive numbers: 7 million active users, 6000 native applications, more than 300,000 active developers, and 100 major new features in Panther, a.k.a. Mac OS X 10.3, the new version of OS X that everybody came to see. Steve then launched into an extensive demo of some ("we don't have time to show them all") of the new Panther goodies.

The most visible new Panther features are in the Finder. Haters of brushed metal will have issues with the Panther Finder, although loyal Mac OS 9 folks will find a few old favorites restored. The mantra for the Finder's new features is "user-centric, not computer-centric". To that end, Apple has added a customizable column of common icons at the left side of Finder windows, including your disks, home folder, and others (see **Figure 2**). This feature is sufficiently raw enough that Apple's department of cool marketing names had not yet taken a firm hold of it: Jobs referred to it as "the left column", although Apple's web site refers to it as the Places sidebar. It gives the Finder a look that's reminiscent of iTunes and the other iApps, which is no doubt intentional.



Figure 2. The new Finder includes a column of icons at the left.

Speaking of iTunes, another cool new Finder trick is interactively winnowing a list of files, iTunes-style. The search

box in the upper-right corner of Finder windows lets you quickly narrow the list of files you're looking at. This handy feature also shows up in the file browser for Xcode, Apple's nifty new developer environment.

The new Action button in the toolbar shows that Apple understands the power of contextual menus and the problems of discoverability. Clicking this button brings up a list of actions that varies depending on the selected object, and you can get the very same list by control-clicking (or right-clicking, because "some people have two-button mice", as Steve acknowledged) on the object.

Panther includes new and improved Open and Save panels. The Places sidebar appears here too, which gives you easy access to your disks, including network volumes. You can look at your files in a list view or column view. See **Figure 3** to take a look at the new Save panel. Another much-requested feature from OS 9, file and folder labels, is back in Panther.



Figure 3. Panther Save panel

One of the coolest bits of iCandy introduced by Apple is a new feature called Exposé. Hmm, Exposé, Rendezvous...those French guys at Apple are infiltrating the department of cool marketing names! Exposé is a solution to the problem of finding the window you want amongst all the windows on the screen. Exposé zooms all your windows down to miniature size and arranges them on the screen so you can see and pick the one you want. Because it's from Apple, Exposé works with flair, smoothly animating all the window-moving business using Quartz Extreme. See **Figure 4** for an idea of how this looks, but you'll have to actually watch it in action to get the full thrilling effect.



Figure 4. Exposé shows all your windows at once, in miniature.

After showing off the Finder, Jobs introduced iChat AV, a new version of the instant message program, now featuring voice and video chatting, or as Jobs called it, "videoconferencing for the rest of us". To show off iChat AV, Steve video-chatted with a friend in Paris and even took a call from Apple board member and former U.S. Vice President Al

Gore. When told he was the second person ever to appear in a live iChat AV chat, Gore looked disappointed and said "Being first is difficult". Apple also shipped a camera, called iSight, to use with the new software.

The new Font Book feature adds lots of power to font management in Panther. You can search for fonts by family and typeface, group font effects, and customize the set of fonts you can see in an application (see **Figure 5**).



Figure 5. Font Book lets you see and manage fonts in Panther.

Here are some of the other new features in Panther that Apple has talked about so far:

Your ONE-STOP Price Comparison for Movies!

MOVIE DEPOT sm

www.moviedepot.com

A new version of Preview that's much faster than the one in Jaguar. Jobs demonstrated very fast scrolling through an enormous PDF doc: the PDF spec itself. Self-reference: whoa! This new version also performs searches much faster than before.

Every print panel now has a Fax button built in. That's very cool, although probably less so to companies who make fax software for OS X.

There's a new edition of Mail that's faster (of course!), lets you drag and drop addresses around when addressing a message, handles threads, and uses Safari's rendering engine to display HTML.

Fast User Switching, which according to Steve is "the only feature we've copied from Windows", lets you quickly change users without having to log out or even quit apps. For the *pièce de résistance* (I'm trying out this French stuff too), Fast User Switching employs a supercool graphic trick, borrowed from Keynote: the old-to-new user transition appears as faces on a rotating cube.

iDisk now lets you work on a local copy of files, then automatically syncs them back to your iDisk. This is especially handy if you work on iDisk files from more than one Mac, as your latest changes will always be available.

New security features include FileVault, which automatically puts 128-bit encryption on files in your home directory, and Secure Empty Trash, which writes random data where your files used to be when you empty the trash.

One of the few downers during Steve's speech came when he said that Panther would be available "before the end of the year", which sounds later than many folks had been guessing. However, Apple shipped a developer preview release at WWDC, which should be available to Select and Premier members of the Apple Developer Connection.

NEWS YOU CAN REALLY USE

After finishing the OS X Panther stuff, Steve introduced Xcode, Apple's new development environment. Xcode is the successor to Project Builder, and it's made for speed and productivity. Xcode includes a new gcc 3.3 compiler that closes much of the speed gap it used to suffer from when compared to CodeWarrior: now, instead of being 10 times slower when compiling the Finder, it's less than twice as slow, which definitely counts as progress.

Jobs described Xcode's distributed builds feature, which uses Rendezvous to find idle computing resources on the local net and puts them to work building big projects in order to speed up project turnaround time. That's pretty cool.

Next, Steve talked about how Xcode removes the linking phase for debug builds, improving turnaround time while you're in the code-build-crash-debug cycle of development. At this point, one clever nerd shouted out "now take out debugging!". Now what fun would that be? In another attempt to speed up your life, Xcode precompiles your headers while you're still typing in the rest of your code, which can save lots of compile time.

After announcing Xcode, Steve called up Chris Espinosa, who has worked at Apple for more than *half his life* (though not just on this project) to show how it works. Chris demonstrated an Xcode feature called Fix & Continue that lets you change an application while it's running, then see the change in the running app. While not quite fully interactive, Fix & Continue is handy for making various edits and changes and then viewing the results immediately. Chris showed a flower-drawing app that was only displaying one petal per flower, which is hardly pretty enough for an Apple demo. Using Fix & Continue, he changed the number of petals per flower (a constant) and we could see the result almost instantly, without having to stop and explicitly rebuild the app. Chris then upped the ante by changing the constant to a function call, and Fix & Continue worked just as well.

Finally, Chris showed the now-familiar search box (just like the ones in iTunes and the Panther Finder) as a way to instantly search for files in your Xcode project. Many long-time Mac developers have never been able to warm to Project Builder, due to a number of issues that include poor performance, tricky user interface, and inertia. Xcode is clearly designed to make those developers think again about switching to Apple's tools.

Watch for much more detail about Xcode in Dave Mark's column next issue.

AS ALWAYS, ONE MORE THING

With the Panther and Xcode announcements successfully out of the way, Jobs turned to his trademark "One more thing..." schtick to issue the day's biggest news: the PowerMac G5, which Apple calls "the world fastest personal computer". Jobs acknowledged that G5 specs had been mistakenly leaked onto Apple's own web site days earlier, then nicely spun the leak by pointing out that although some thought the specs were too good to be true, they really were accurate after all.

Jobs described the partnership with IBM that led up to the creation of the 64-bit G5 chip, which IBM calls PowerPC 970. The raw specs are pretty cool: up to 2 GHz clock speed, with 3 GHz promised within a year; 1 GHz front-side bus; massively parallel architecture that can have an incredible 215 instructions in flight at once; dual floating point units; and a super-duper dual pipeline Velocity Engine. Jobs brought out IBM Vice President John Kelly, who looked only slightly out of place at an Apple shindig (jacket, no tie) and talked about the \$3 billion factory that IBM uses to make these puppies.

After announcing the G5 chip itself, Steve described the system Apple built to go around it. This includes a clever system controller that speeds things up in a number of ways, including dedicated bandwidth for main memory, fast PCI and PCI-X slots, and huge expansion, including up to 8 Gb RAM.

With the chip and geeky system details described, the new Power Mac G5 arose like a golden calf from a previously hidden pedestal on the stage. The aluminum design is boxier than previous Power Macs, and features lots of little holes to help



Peachpit

Essential books for the creative community

Beat the Heat!

Don't sweat it. Jack up the AC and chill out with these cool, new summer releases designed to help you make the most of your Mac.

The Little Mac iApps Book

By John Tollett with Robin Williams
0-321-18747-4 • \$21.99

The Mac OS X Conversion Kit: 9 to 10 Side by Side

By Scott Kelby
0-7357-1354-5 • \$29.99

The Macintosh iLife

By Jim Heid
0-321-17011-3 • \$29.99

Final Cut Express for Mac OS X: Visual QuickStart Guide

By Lisa Brenneis
0-321-19912-X • \$21.99

Apple Pro Training Series: Shake 3

By Marco Paolini
0-321-19725-9 • \$44.99

Making Movies with Final Cut Express

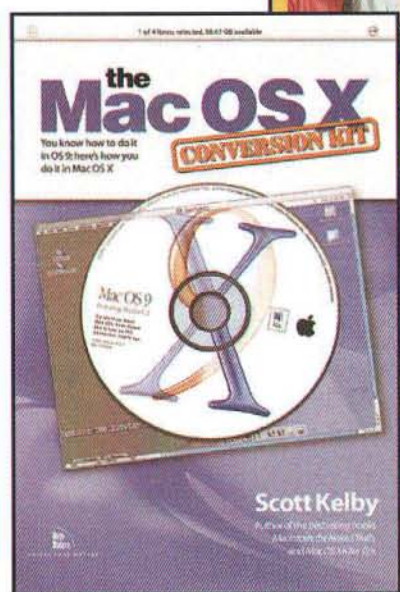
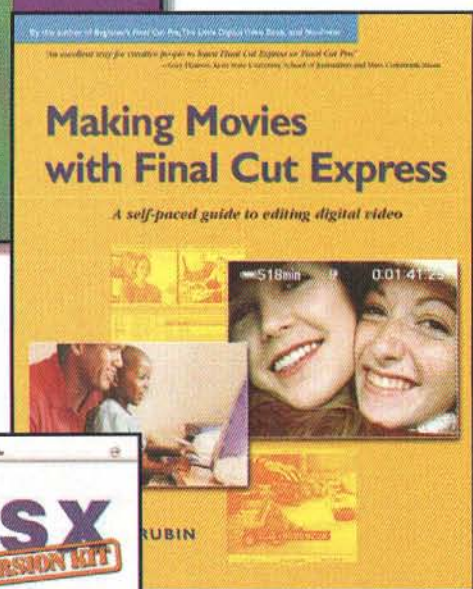
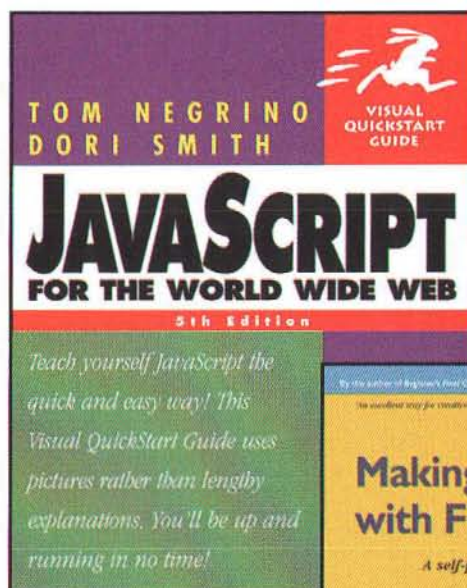
By Michael Rubin
0-321-19777-1 • \$29.99

Mac OS X Security

By Bruce Potter, Preston Norvell and Brian Wotring
0-7357-1348-0 • \$39.99

JavaScript for the World Wide Web, Fifth Edition: Visual QuickStart Guide

By Tom Negrino and Dori Smith
0-321-19439-X • \$21.99



Save up to 30%! Become a Peachpit Club Member today!

Enjoy 10% off all books every day at peachpit.com, earn an additional 10% discount as a Peachpit Club Member, and save 10% on top of that with this one-time coupon! Simply go to www.peachpit.com/mactech703 and enter coupon code EM-73AA-MTMF at checkout. It's that easy!



Peachpit Press

New
Riders

Whistle Blower

Enterprise server monitor and restart utility
whistleblower.sentman.com

Connect to and validate the response from web servers, cgi scripts and over 23 other types of servers.

Send email, pages and perform unattended restarts via MasterSwitch or PowerKey.

Shifts make sure that the person on call when the server goes down is the one who gets the page.

68k, PPC and Carbon

Web based administration lets you check on and restart your servers from anywhere.

Customize your response to an outage with Apple Script.

email us at whistleblower@sentman.com

ventilate those heat-producing G5s (see **Figure 6**). In fact, the G5 has *nine* fans for cooling, an announcement that brought a groan from the audience that has suffered with "wind-tunnel" G4s. Steve explained that the fan armada is controlled by software, not just dumb sensors, which will actually lead to quieter computers, but many in the crowd seemed to be skeptical and adopted a wait-and-see (or, more accurately, wait-and-hear) attitude.

MacTech[®]
M A G A Z I N E

Get MacTech delivered to your door at a price
FAR BELOW the newsstand price.
And, it's **RISK FREE!**
Subscribe Today!
www.mactech.com

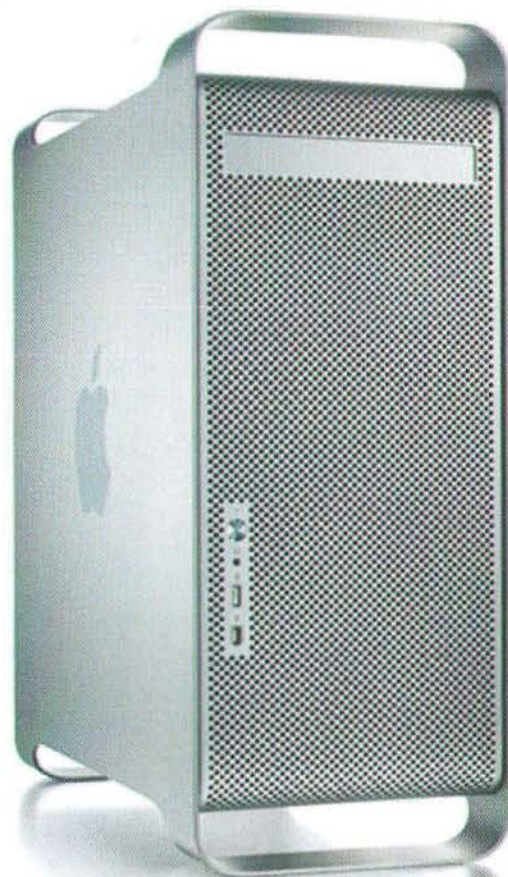


Figure 6. Power Mac G5. This is the front, which includes several ports for easy access.

The coolest parts of Steve's G5 demo were the "bake-offs" and charts that compared its speed to various Pentium and Zeon configurations. This time, the claims included a new wrinkle: an independent testing house that verified Apples "fastest personal computer" boast. Those claims were much debated following the keynote: one key question was whether the tests used the best C compiler for the Intel machines. But nit-picking aside, it's clear that for the first time in a long while, the PowerMac G5 finally restores Apple to a competitive position in raw system speed.

STEVE IS REALLY JUST LIKE US

After the keynote, I returned to the high tech bathroom facilities. As I was drying my hands at the sink, who should bound into view but the man himself, Steve Jobs. I said hi, he said hi, and I walked out of the bathroom as he stood at the sink. As I walked out, I heard the whirr of the automatic paper towel dispenser, followed by the sound of Steve chuckling in amusement. So here's the early rumor: look for Apple to ship automated paper towel dispensing hardware in every PowerMac G6.



Keep track of every second of your time and bill for it with Time Track! Built in instructions make it easy to use. It is a simple way to manage your billable time for multiple projects and create a web page to show to your clients. Only \$24.95 per single user license per platform. Finally! A versatile time tracking solution for Macintosh, Windows, and Palm!

www.trinfinitysoftware.com



By TLA Systems

The Dock with more than one dimension.

Create multiple docks of any size, assign hot keys and even put the Trash back on the Desktop. A flexible and feature-laden tool for power-users. Runs natively on Mac OS X and 9 in five languages.

"...you made the switch to OS X a lot easier for me..." - Bob LeVitus

"...DragThing can rightfully be called an indispensable aid to working with your Mac..." - MacUser UK

Download a copy now from www.dragthing.com.

The Journal of Macintosh Technology & Development

MacTech

Got a great product sold through Kagi?

Promote your product through the very cost effective Kagi Showcase in MacTech Magazine.

Contact us today for special rates!

adsales@mactech.com



Eudora Internet Mail Server (EIMS) 3.1 is the latest version of the most popular Internet mail server for the Macintosh. If you need to handle email for a dozen users, or thousands of users, EIMS is a reliable and easy to use solution.

EIMS 3.1 is available for US\$400.00, there are no limits on the number of users that can be added, and free email support is included.

For more information, see

<http://www.eudora.co.nz/>



GraphicConverter converts pictures to different formats. Also it contains many useful features for picture manipulation.

See **www.lemkesoft.com** for more information.

List of Advertisers

Aladdin Knowledge Systems, Inc.	11
Aladdin Systems, Inc.	21
Big Nerd Ranch, Inc.	59
Brad Sniderman	34
Circus Ponies Software, Inc.	45
DevDepot	28-29
digitalforest	35
EazyDraw (Dekorrra Optics, LLC)	68
effigent, Inc.	55
Electric Butterfly	40
Eudora Internet Mail Server	79
Excel Software	19
FairCom Corporation	1
Felt Tip Software	18
Fetch Softworks	9
Full Spectrum Software, Inc.	40
IDG World Expo Corporation	50-51
James Sentman Software	78
Jiiva, Inc.	56
Lemke Software GmbH	79
Lingo Systems	49
MacDirectory	7
MacTech Magazine	73
Mathemaesthetics, Inc.	43
MCF Software	23
Movie Depot	75
MYOB US, Inc.	25
/n software inc.	39
Netopia, Inc.	63
Paradigma Software	27
Parliant Corporation	61
Peachpit Press	77
Perforce Software, Inc.	81
piDog Software	54
PowerGlot Software	57
Presto Vivace, Inc.	22
PrimeBase (SNAP Innovation)	47
Prosoft Engineering, Inc.	65
Redstone Software, Inc.	53
Runtime Revolution Limited	82
Seapine Software, Inc.	71
Small Dog Electronics	69
Sophisticated Circuits, Inc.	41
Sybase, Inc.	2-3
ThinkFree Corporation	17
TLA Systems Ltd.	79
Trango Broadband Wireless	15
Trinifinity Software	79
Trolltech AS	IFC
UNI SOFTWARE PLUS GMBH	37
Utilities4Less.com	33
VVI	67
WIBU-SYSTEMS AG	13

List of Products

Adobe Press • Peachpit Press	77
Big Nerd Ranch • Big Nerd Ranch, Inc.	59
c-tree Plus • FairCom Corporation	1
Development & Testing • Full Spectrum Software, Inc.	40
Disk Utilities • Prosoft Engineering, Inc.	65
DragThing • TLA Systems Ltd.	79
Eazy Draw • EazyDraw (Dekorrra Optics, LLC)	68
Eggplant • Redstone Software, Inc.	53
EIMS • Eudora Internet Mail Server	79
Enterprise Software • MCF Software	23
Fetch • Fetch Softworks	9
Graphic Converter • Lemke Software GmbH	79
Hosting Solutions • digitalforest	35
InstallerMaker, StuffIt • Aladdin Systems, Inc.	21
IPWorks! • /n software inc.	39
Law Offices • Brad Sniderman	34
Long Distance Phone Service • Utilities4Less.com	33
MacDirectory • MacDirectory	7
MacTech Magazine Subscription • MacTech Magazine	73
Macworld Conference & Expo • IDG World Expo Corporation	50-51
Maximizing Your Mac! • DevDepot	28-29
moviedepot.com • Movie Depot	75
MYOB • MYOB US, Inc.	25
Notebook • Circus Ponies Software, Inc.	45
Phone Valet • Parliant Corporation	61
piDog Utilities • piDog Software	54
PowerGlot • PowerGlot Software	57
PowerKey Pro & KickOff! • Sophisticated Circuits, Inc.	41
Presto Vivace Services • Presto Vivace, Inc.	22
PrimeBase • PrimeBase (SNAP Innovation)	47
Qt • Trolltech AS	IFC
Quick UML for Mac OS X • Excel Software	19
Resorcerer • Mathemaesthetics, Inc.	43
Revolution • Runtime Revolution Limited	82
SCM Software • Perforce Software, Inc.	81
Security • Aladdin Knowledge Systems, Inc.	11
SmallDog.com • Small Dog Electronics	69
Software Protection • WIBU-SYSTEMS AG	13
Sound Studio • Felt Tip Software	18
SuperScrubber & Application Builder Collection • Jiiva, Inc.	56
Sybase services • effigent, Inc.	55
Sybase • Sybase, Inc.	2-3
TestTrack Pro • Seapine Software, Inc.	71
ThinkFree • ThinkFree Corporation	17
Timbuktu Pro & netOctopus • Netopia, Inc.	63
Time Track • Trinifinity Software	79
Translation & Localization • Lingo Systems	49
UniHelp Module • Electric Butterfly	40
Valentina • Paradigma Software	27
Visual-Report Tool Developer • VVI	67
VOODOO Server • UNI SOFTWARE PLUS GMBH	37
WhistleBlower • James Sentman Software	78
Wireless Networking • Trango Broadband Wireless	15

The index on this page is provided as a service to our readers. The publisher does not assume any liability for errors or omissions.

Want to connect your development teams seamlessly?
You've got two choices:

Use Perforce **or** Spin your wheels



Perforce is the Fast Software Configuration Management System that gets distributed teams working together in no time.

Other systems require an army of consultants, months of hacking, and probably the odd dab of crazy glue. With Perforce, nationwide and worldwide development teams can focus on working together without worrying about holding it together.

Work anywhere. Requiring only TCP/IP, Perforce makes use of a well-tuned streaming message protocol for synchronizing client workspaces with server repository contents. And the Perforce Proxy offers caching to provide remote users with quick download times while maintaining real-time access to project activity and status information.

With Perforce, the fast way is the right way. Install it fast, learn it fast, run it fast. With other SCM systems, developers face an unpleasant choice: do it the right way or do it the fast way. Perforce's speed and reach mean the fast way is the right way. See how Perforce compares with other leading SCM systems at: <http://www.perforce.com/perforce/reviews.html>

Run at full speed even with hundreds of users and millions of files. At the core of Perforce lies a relational database with well-keyed tables, so normal user activity can be accomplished in near-zero time. Larger operations (like labeling a release and branching) are translated into keyed data access, giving Perforce the scalability that big projects require.

Develop and maintain multiple codelines. Perforce's Inter-File Branching lets you merge new features and apply fixes across codelines. Smart metadata keeps track of your evolving projects even while they are developed in parallel.

Truly cross platform. Perforce runs on more than 50 operating systems, including Windows®, UNIX®, Linux®, Mac OS® X, AS/400, and more.

Integrate with leading IDEs and defect trackers: Visual Studio.NET®, Visual C++®, Visual Basic®, JBuilder®, CodeWarrior®, TeamTrack®, Bugzilla™, ControlCenter®, DevTrack® packages, plus more.

PERFORCE
SOFTWARE

Fast Software Configuration Management www.perforce.com

Download your free 2-user, non-expiring, full-featured copy now from www.perforce.com
Free (and friendly) technical support is on hand to answer evaluation questions.

All trademarks used herein are either the trademarks or registered trademarks of their respective owners.

It'll Give You A Life.

Nassau Beach, Bahamas - Sun, sand and soft breezes make this one of the most relaxing vacation spots in the world. Don't forget the cool, fruity drink!

Dear Revolution 2.0,

Thanks for the
great vacation!

Missing you terribly,

Love,

Cecil



Revolution 2.0
c/o My Computer 64.23.0.192
24-7 Relief From Aggravation
(formerly DullAndDreary Coders)
The Corporate World, #1-EASY

But A Geek Is Always A Geek.

**Revolution 2.0: the English like language designed around the way you think.
Develop and deliver on Mac OS X, Windows and Linux
(not to mention 10 other platforms).**

Take the English language, add elegant XML, more SQL databases than you care to learn, intrinsic video capture, Unicode, CGI scripting, sophisticated Reports, and build your solution faster than anyone else. Jaguar, Panther, XP? Revolution lets you eat platforms for breakfast. And speaking of eating, our Cookbook of examples gets you up, running, and productive NOW. You can join the Revolution for as little as \$99...
Build with it, deliver with it, love it - and still have time for vacations.

And for a limited time, your MacTech special lets you get 100% of the Revolution for 15% off - go to special.runrev.com NOW. Thousands of developers have already joined. Don't let the Revolution in modern coding start without you...

Software At The Speed Of Thought



Runtime Revolution • 91 Hanover Street • Edinburgh EH2 1DJ • UK
Phone +44 (0)131 718 4333 • Fax +44 (0)131 718 4334 • www.runrev.com • Email info@runrev.com